AD-A031 440 GENERAL DYNAMICS FORT WORTH TEX CONVAIR AEROSPACE DIV F/G 17/9
ENDO ATMOSPHERIC EXO ATMOSPHERIC RADAR MODELING. RADAR CROSS SE--ETC(U)
JUN 76 R J HANCOCK, F H CLEVELAND F30602-73-C-0380

UNCLASSIFIED RADC-TR-76-186-VOL-4-PT-2 NL

1 OF 4
AD
A031440

FC (12)

RADC-TR-76-186, Vol IV, Pt 2
Final Technical Report
June 1976

ENDO ATMOSPHERIC-EXO ATMOSPHERIC RADAR MODELING
(Appendix)

General Dynamics

COPY AVAILABLE TO DDC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION

Approved for public release;
distribution unlimited.

Laboratory Directors' Fund Number 01707310
with Proj. 6512 funds added.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

APPROVED: John C. Cleary
JOHN C. CLEARY
Project Engineer

APPROVED: Moses A. Diab
MOSES A. DIAB
Acting Technical Director
Surveillance Division

FOR THE COMMANDER: John P. Huss
JOHN P. HUSS
Acting Chief, Plans Office

Initially funded under RADC Laboratory Directors' Funds with Proj. 6512 funds added.

19 TR-76-186-Vol-4-Pt-2

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
18 RADC-TR-76-186, Vol IV, Pt 2

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
6 ENDO ATMOSPHERIC-EXO ATMOSPHERIC RADAR MODELING.
Radar Cross Section Modeling.
Appendix L.

5. TYPE OF REPORT & PERIOD COVERED
9 Final Technical Report.

6. PERFORMING ORG. REPORT NUMBER
N/A

7. AUTHOR(s)
10 Robert J. Hancock
Fred H. Cleveland

8. CONTRACT OR GRANT NUMBER(s)
15 F30602-73-C-0380

9. PERFORMING ORGANIZATION NAME AND ADDRESS
General Dynamics
Convair Aerospace Division
P O Box 748, Ft Worth TX 76101

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61101F
LDF 01707310

11. CONTROLLING OFFICE NAME AND ADDRESS
Rome Air Development Center (OCSA)
Griffiss AFB NY 13441

12. REPORT DATE
11 June 1976

13. NUMBER OF PAGES
418
12 312 p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Same

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
Same

18. SUPPLEMENTARY NOTES
RADC Project Engineer: John C. Cleary (OCSA)
Initially funded under RADC Laboratory Directors' Funds with Proj. 6512 funds added.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Radar
Simulation
Computer Modeling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This effort is concerned with the development and implementation of a set of digital computer programs that will augment the RADC digital computer radar simulation model procured under Contr F30602-72-C-0393 (01707201). The computer programs shall consist of a sequence of subroutines that correspond to separate functions such as a chaff model, target model, propagation effects and clutter model. The original radar simulation model will be expanded to include a bistatic capability and will include ECM and phase coded pulse compression re-

over

406 220

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

bpg

ceiver techniques. In addition, an interactive syscem has been designed for the simulation. Using an interactive display, an engineer would be able to understand what is happening by being able to observe results at several intermediate points in the problem. A picture is worth a thousand words. For example, an antenna pattern or waveform response to a target is more meaningful than a long table of numerical listings. Parts of the simulation were used by RADC for Deep Space Surveillance Radar (DSSR) waveform analysis, generating antenna patterns and tradeoffs involving phase shifter bit-size for the Advanced Space Defense Program (ASDP). The RADC radar simulation model is being used to support Seek Sail, Cobra Judy, Digital Coded Radar and Seek Sentry.

This report contains a large percentage of machine-produced copy which is not of the highest printing quality but because of economical consideration, it was determined in the best interest of the government that they be used in this publication.

This report contains Vol I, Pt 1 (Sections 1-7 and 9-10) (Pages 1-1 thru 1-5, 2-1 thru 2-24, 3-1 thru 3-35, 4-1 thru 4-23, 5-1 thru 5-6, 6-1 thru 6-39, 7-1 thru 7-30, 9-1 thru 9-3 and 10-1 thru 10-2).

Vol I, Pt 2 contains Section 8 (Pages 8-1 thru 8-174).

Vol I, Pt 3 contains Section 8 (Pages 8-175 thru 8-418).

Vol II, Pt 1 contains(Sections 1-8 and 10 & 11) (Pages 1-1, 2-1 thru 2-24, 3-1 thru 3-15, 4-1 thru 4-137, 5-1 thru 5-16, 6-1 thru 6-44, 7-1, 8-1 thru 8-26, 10-1 thru 10-4 and 11-1 thru 11-2).

Vol II, Pt 2 contains Sections 9 and 10 (Pages 9-1 thru 9-234 and Pages 10-1 thru 10-4).

Vol III contains Sections 1 thru 6 (Pages 1-1 thru 1-2, 2-1 thru 2-22, 3-1 thru 3-53, 4-1 thru 4-141, 5-1 thru 5-3 and 6-1).

Vol IV, Pt 1 contains Appendices A-K and Appendix M.

Vol IV, Pt 2 contains Appendix L.

TABLE OF CONTENTS

APPENDIX TITLE PAGE

LIST OF FIGURES

APPENDIX L

RADAR CROSS SECTION

ANALYTIC MODEL DESCRIPTIONS

The FORTRAN IV computer listings for each of the radar cross section models are located at the end of each of the appropriate sections L.1 through L.7. Although the programs have not been integrated into the Radar Simulation Model, they can be easily made to interface with RADSIM. In the interim they can be run as independent FORTRAN subroutines, which are stored on disc file on the RADC HIS-635 computer under USERID-CLEARY and account number 017073100380 in the following files:*

SECTION	NAME	SUB-FILE	LINES
L.1	Cylinder with Spheroidal Caps	RCSM3	10760, 11430
L.2	Hemisphere-Cylinder	RCSM3	11440, 13680
L.3	Cone, Cylinder or Frustrum	RCSM3	13690, 17460
L.4	Thin Wire	RCSM1	50, 3120
L.5	Frustrum-Cylinder-Frustrum	RCSM1	3130, 7580
L.6	Cylinder-Frustrum Combination	RCSM1	7590, 12170
L.7	Stepped Cylinder	RCSM2	50, 3710

The outputs for all programs consist of four linear arrays - EVVR, EVVI, EHHR and EHHI - which contain the real and imaginary parts of the vertically and horizontally polarized back scattered fields as a function of frequency. The fields are computed at discrete frequency increments (DF) in megahertz, between user defined limits. The low and high frequency limits are equal to (NMIN-1)*DF & (NMAX-1)*DF respectively, where NMIN and NMAX are dimensionless indices. The magnitudes of the arrays - EVVR, EVVI, EHHR and EHHI - have been scaled in order that the target cross section (in square meters) can be obtained from the sum of the squares of the corresponding real and imaginary values of the array.

* A typical use of SUBROUTINE TARGET is shown in Volume I, Part 3, Section 8.24. All subroutines in this appendix can be run using this same FORTRAN program, labelled RCSSP.

L.1 CYLINDER WITH SPHEROIDAL CAPS (RADCAT)

The far-field scattering from the RADCAT vehicle, a large cylinder with spheroidal end caps, has been formulated using an expression involving a physical optics approximation of the backscattering from a cylindrical section and a geometrical optics approximation to the scattering from a spheroid (Ref 1). The resulting expression of the scattering field is the following:

$$e(\theta) = \ell \sqrt{ka\sin\theta} \left\{\frac{\sin(k\ell\cos\theta)}{k\ell\cos\theta}\right\} e^{-i(2ka\sin\theta + \pi/4)}$$

$$+ \left\{\frac{\sqrt{\pi}\, a^2 c}{a^2 \sin^2\theta + c^2 \cos^2\theta}\right\} e^{-i2k(\frac{\ell}{2}\cos\theta+\sqrt{a^2\sin^2\theta+c^2\cos^2\theta})}$$

where the physical dimensions of the target and target geometry are shown in Figure L.1-1 and k = wave number = $2\pi/\lambda$.

The computer program subroutine is used to compute the field backscattered from the target for the case of vertical and horizontal polarizations with respect to the target rotational plane (defined by the target axis of symmetry and the radar line of sight). It should be noted that for the case of this particular target, the formulation of the scattered field is polarized insensitive. The inputs, outputs, restrictions, and definition of key terms of this subroutine are presented in the following paragraphs.

L.1.1 Inputs

The subroutine inputs are read from cards or passed in a common block. The used inputs passed in the common block are the following:

FC = Carrier Frequency in GHz, not necessarily the mid frequency

DF = Frequency increment in MHz

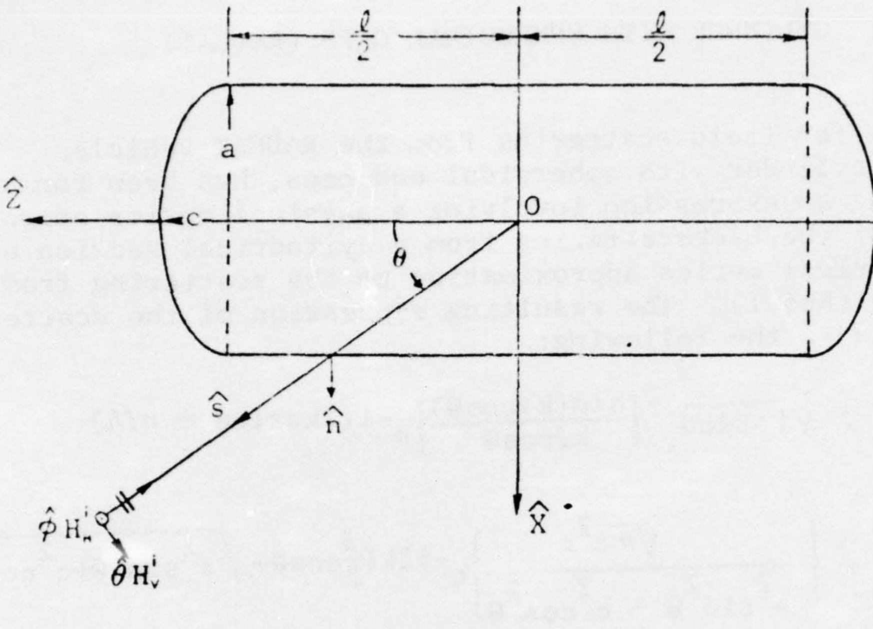


Fig. L.1-1 RADCAT TARGET GEOMETRY

NMIN = Minimum frequency index, i.e. f_{min} = DF*(NMIN-1)

NMAX = Maximum frequency index, i.e. f_{max} = DF*(NMAX-1)

The inputs which are read from a card are the following:

	MATH SYMBOL	VARIABLE	DEFINITION	COLUMNS
Card 1	a	XA	Radius of cylindrical section (feet)	1-6
	c	XC	Depth of Spheroidal Cap (feet)	7-12
	ℓ	XL	Length of cylindri-section (feet)	13-18
	θ	THETA	Azimuth angle (degrees)	19-24

L.1.2 Outputs

The outputs consist of four linear arrays EVVR, EVVI, EHHR, EHHI which contain the real and imaginary parts of the vertically and horizontally polarized back scattered fields (in meters) at frequency increments of DF between DF*(NMIN-1) and DF*(NMAX-1). The bandwidth of the output then equals DF*(NMAX-NMIN).

L.1.3 Restrictions

L.1.3.1 Physical Dimensions

As high-frequency approximations were utilized, all physical target dimensions should be large with respect to the wavelength of the illuminating field. Thus, XA, XC, and XL should be larger than the wavelength of the smallest frequency at which a computation is obtained. The condition XC < XA is also implied in the formulation of the problem.*

L.1.3.2 Output

The output arrays are passed in the argument list and a value is computed only for array locations from NMIN to NMAX.

*Also, $1 \leq \text{NMIN} \leq \text{NMAX} \leq 512$.

L.1.3.3 Azimuth

The azimuth angle is restricted to the region between 0 and 90 degrees.

L.1.4 Definition of Key Terms Used in Subroutine

$$\text{TERM 7} = + \left| \frac{\sqrt{\Pi}\quad a^2 c}{a^2 \sin^2 \theta + c^2 \cos^2 \theta} \right|$$

$$\text{TERM 2} = \ell\sqrt{ka\sin\theta}$$

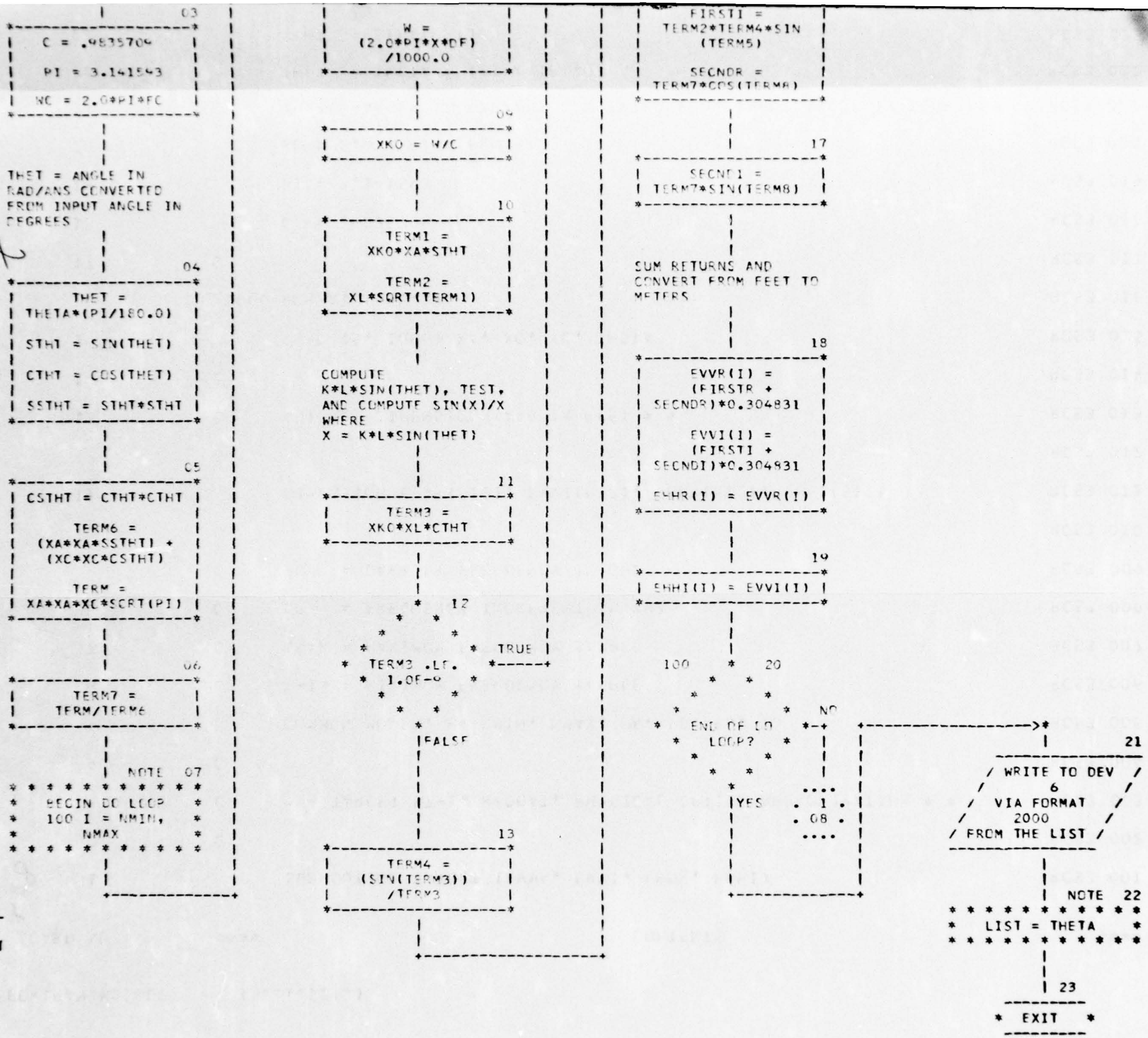
$$\text{TERM 4} = \frac{\sin(k\ell\cos\theta)}{k\ell\cos\theta}$$

$$\text{TERM 5} = (2ka\sin\theta + \Pi/4)$$

$$\text{TERM 8} = 2k(\frac{\ell}{2}\cos\theta + \sqrt{a^2\sin^2\theta + c^2\cos^2\theta}$$

```
$       IDENT   BECAGD01,HANCOKC,017073100380,DSTOR2
$       OPTION  FORTRAN
$       FORTY   LSTIN,XREF,MAP,DECK
$       LIMITS  05,39K,0,5K
      SUBROUTINE TARGET (EVVR, EVVI, EHHR, EHHI)
C
C     * * TARGET ST-1, RADCAT, PHYSICAL 0PTICS APPROXIMATION * *
C
      COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, T0
C     NMIN = MINIMUM FREQUENCY SAMPLE
C     NMAX = MAXIMUM FREQUENCY SAMPLE
C     DF   = FREQUENCY INCREMENT IN MHZ
C     FC   = CARRIER FREQUENCY IN GHZ
C
      DIMENSION EVVR(512), EVVI(512), EHHR(512), EHHI(512)
C
C     * * ALL DIMENSIONS ARE IN FEET * *
C
      READ (5, 1000) XA, XC, XL, THETA
 1000 FORMAT (4F6.2)
C
      C  = .9835703
      PI = 3.141593
      WC = 2.0 * PI * FC
C     THET = ANGLE IN RAD/ANS CONVERTED FROM INPUT ANGLE IN DEGREES
      THET  = THETA * (PI / 180.0)
      STHT  = SIN(THET)
      CTHT  = COS(THET)
      SSTHT = STHT * STHT
      CSTHT = CTHT * CTHT
      TERM6 = (XA * XA * SSTHT) + (XC * XC * CSTHT)
      TERM  = XA * XA * XC * SORT(PI)
      TERM7 = TERM / TERM6
C
      DO 100 I=NMIN, NMAX
      X  = (I-1)
      W = (2.0 * PI * X * DF) / 1000.0
C
      XK0 = W / C
C
      TERM1 = XK0 * XA * STHT
      TERM2 = XL * SORT(TERM1)
C     COMPUTE K*L*SIN(THET), TEST, AND COMPUTE SIN(X)/X WHERE
C             X = K*L*SIN(THET)
      TERM3 = XK0 * XL * CTHT
      IF (TERM3 .LE. 1.0E-9) GO TO 10
      TERM4 = (SIN(TERM3)) / TERM3
      GO TO 20
   10 TERM4 = 1.0
C     COMPUTE PHASE TERM FOR CYLINDER(TERM5) AND SPHEROID(TERM8)
   20 TERM5 = (2.0 * TERM1) + (PI / 4.0)
      TERM8 = TERM3 + (2.0 * XK0 * SQRT(TERM6))
C     COMPUTE REAL AND IMAGINARY PARTS OF CYLINDRICAL (FIRSTR,FIRSTI)
C             AND OBLATE SPHEROIDAL (SECNDR,SECNDI) RETURNS
      FIRSTR = TERM2 * TERM4 * COS(TERM5)
      FIRSTI = TERM2 * TERM4 * SIN(TERM5)
      SECNDR = TERM7 * COS(TERM8)
      SECNDI = TERM7 * SIN(TERM8)


76    16 08


C
C     SUM RETURNS AND CONVERT FROM FEET TO METERS
      EVVR(I) = (FIRSTR + SECNDR) * 0.304831
      EVVI(I) = (FIRSTI + SECNDI) * 0.304831
      EHHR(I) = EVVR(I)
      EHHI(I) = EVVI(I)
  100 CONTINUE
C
      WRITE (6, 2000) THETA
 2000 FORMAT ('      ASPECT ANGLE = ', F6.2, ' DEG')
C
      RETURN
C
      END
```

```
04/26/76                                                AUTOFLOW CHART SET - FWD/SCL  RADSIM

CHART TITLE - SUBROUTINE  TARGET(EVVR,EVVI,EHHR,EHHI)


      ----------                                                          -----------
      / TARGET /                                                          /   10    /
      ----------                                                          -----------
                                                                              |
           *                                                +---------------->|
                                                            |                 |        14
 * * TARGET ST-1,                                           |        *--------------------*
RADCAT, PHYSICAL                                            |        |    TERM4 = 1.0     |
OPTICS                                                      |        *--------------------*
APPROXIMATION * *                                           |                 |
                                                            |                 |
                                                            |                 |
                                                            |        COMPUTE PHASE TERM
NMIN = MINIMUM                                              |        FOR CYLINDER(TERM5)
FREQUENCY SAMPLE                                            |        AND SPHEROID(TERM8)
NMAX = MAXIMUM                                              |                 |
FREQUENCY SAMPLE                                            |     +---------->|
DF    = FREQUENCY                                           |     |     20    |        15
INCREMENT IN MHZ                                            |     |  *--------------------*
FC    = CARRIER                                             |     |  |      TERM5 =       |
FREQUENCY IN GHZ                                            |     |  |   (2.0*TERM1) +    |
                                                            |     |  |      (PI/4.0)      |
                                                            |     |  |                    |
                                                            |     |  |  TERM8 = TERM3 +   |
 * * ALL DIMENSIONS                                         |     |  |  (2.0*XKO*SQRT     |
ARE IN FEET * *                                             |     |  |     (TERM6))       |
                                                            |     |  *--------------------*
          *                                                 |     |           |
          |        01                                       |     |           |
     ---------------                                        |     |           |
    / READ FROM DEV /                                       |     |  COMPUTE REAL AND
   /       5       /                                        |     |  IMAGINARY PARTS OF
  /   VIA FORMAT  /                                         |     |  CYLINDRICAL
 /      1000     /                                          |     |  (FIRSTR,FIRSTI)
/ INTO THE LIST /                                           |     |  AND OBLATE SPHEROIDAL
---------------                                             |     |  (SECNDR,SECNDI)
          |                                                 |     |  RETURNS
          |                                                 |     |           |
          | NOTE  02                                        |     |           |
* * * * * * * * * * * *                                     |     |           |        16
*  LIST = XA, XC,     *                                     |     |  *--------------------*
*     XL, THETA       *  +-----------01.07*-->*             |     |  |      FIRSTR =      |
* * * * * * * * * * * *  |                    |        08   |     |  |  TERM2*TERM4*COS   |
          |              |         *--------------------*   |     |  |      (TERM5)       |
          |              |         |    X = (I - 1)     |   |     |  |                    |
          |        03    |         |                    |   |     |  |      FIRSTI =      |
*--------------------*   |         |        W =         |   |     |  |  TERM2*TERM4*SIN   |
|   C = .9835709     |   |         |  (2.0*PI*X*DF)     |   |     |  |      (TERM5)       |
|                    |   |         |     /1000.0        |   |     |  |                    |
|   PI = 3.141593    |   |         *--------------------*   |     |  |      SECNDR =      |
|                    |   |                    |             |     |  |  TERM7*COS(TERM8)  |
|   WC = 2.0*PI*FC   |   |                    |             |     |  *--------------------*
```


03
C = .9835704
PI = 3.141593
WC = 2.0*PI*FC
THET = ANGLE IN RAD/ANS CONVERTED FROM INPUT ANGLE IN DEGREES
04
THET = THETA*(PI/180.0)
STHT = SIN(THET)
CTHT = COS(THET)
SSTHT = STHT*STHT
05
CSTHT = CTHT*CTHT
TERM6 = (XA*XA*SSTHT) + (XC*XC*CSTHT)
TERM = XA*XA*XC*SQRT(PI)
06
TERM7 = TERM/TERM6
NOTE 07
BEGIN DO LOOP 100 I = NMIN, NMAX
W = (2.0*PI*X*DF) /1000.0
09
XKO = W/C
10
TERM1 = XKO*XA*STHT
TERM2 = XL*SQRT(TERM1)
COMPUTE K*L*SIN(THET), TEST, AND COMPUTE SIN(X)/X WHERE X = K*L*SIN(THET)
11
TERM3 = XKO*XL*CTHT
12
TERM3 .LE. 1.0E-6
TRUE
FALSE
13
TERM4 = (SIN(TERM3)) /TERM3
FIRSTI = TERM2*TERM4*SIN (TERM5)
SECNDR = TERM7*COS(TERM8)
17
SECNDI = TERM7*SIN(TERM8)
SUM RETURNS AND CONVERT FROM FEET TO METERS
18
EVVR(I) = (FIRSTR + SECNDR)*0.304831
EVVI(I) = (FIRSTI + SECNDI)*0.304831
EHHR(I) = EVVR(I)
19
EHHI(I) = EVVI(I)
100
20
END OF LO LOOP?
NO
YES
08
21
WRITE TO DEV 6 VIA FORMAT 2000 FROM THE LIST
NOTE 22
LIST = THETA
23
EXIT

L-8

```
04/26/76          INPUT LISTING                         AUTOFLOW CHART SET - FWD/SCL  RADSIM

FORTRAN MODULE        (SCC,LIST2)

  CARD NO         ****                          CONTENTS                                  ****

     1                   SUBROUTINE TARGET (EVVR, EVVI, EHHR, EHHI)                   RCS3 001
     2            C                                                                   RCS3 002
     3            C      * * TARGET ST-1, RADCAT, PHYSICAL OPTICS APPROXIMATION * *   RCS3 003
     4            C                                                                   RCS3 004
     5                   COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, TO                  RCS3 005
     6            C      NMIN = MINIMUM FREQUENCY SAMPLE                              RCS3 006
     7            C      NMAX = MAXIMUM FREQUENCY SAMPLE                              RCS3 007
     8            C      DF   = FREQUENCY INCREMENT IN MHZ                            RCS3 008
     9            C      FC   = CARRIER FREQUENCY IN GHZ                              RCS3 009
    10            C                                                                   RCS3 010
    11                   DIMENSION EVVR(512), EVVI(512), EHHR(512), EHHI(512)         RCS3 011
    12            C                                                                   RCS3 012
    13            C      * * ALL DIMENSIONS ARE IN FEET * *                           RCS3 013
    14            C                                                                   RCS3 014
    15                   READ (5, 1000) XA, XC, XL, THETA                             RCS3 015
    16             1000  FORMAT (4F6.2)                                               RCS3 016
    17            C                                                                   RCS3 017
    18                   C  = .9835709                                                RCS3 018
    19                   PI = 3.141593                                                RCS3 019
    20                   WC = 2.0 * PI * FC                                           RCS3 020
    21            C      THET = ANGLE IN RADIANS CONVERTED FROM INPUT ANGLE IN DEGREES RCS3 021
    22                   THET  = THETA * (PI / 180.0)                                 RCS3 022
```

```
      WC = 2.0 * PI * FC                                          RCS3 020
C     THET = ANGLE IN RADIANS CONVERTED FROM INPUT ANGLE IN DEGREES RCS3 021
      THET  = THETA * (PI / 180.0)                                RCS3 022
      STHT  = SIN(THET)                                           RCS3 023
      CTHT  = COS(THET)                                           RCS3 024
      SSTHT = STHT * STHT                                         RCS3 025
      CSTHT = CTHT * CTHT                                         RCS3 026
      TERM6 = (XA * XA * SSTHT) + (XC * XC * CSTHT)               RCS3 027
      TERM  = XA * XA * XC * SQRT(PI)                             RCS3 028
      TERM7 = TERM / TERM6                                        RCS3 029
C                                                                 RCS3 030
      DO 100 I=NMIN, NMAX                                         RCS3 031
      X  = (I-1)                                                  RCS3 032
      W = (2.0 * PI * X * DF) / 1000.0                            RCS3 033
C                                                                 RCS3 034
      XKO = W / C                                                 RCS3 035
C                                                                 RCS3 036
      TERM1 = XKO * XA * STHT                                     RCS3 037
      TERM2 = XL * SQRT(TERM1)                                    RCS3 038
C     COMPUTE K*L*SIN(THET), TEST, AND COMPUTE SIN(X)/X WHERE     RCS3 039
C           X = K*L*SIN(THET)                                     RCS3 040
      TERM3 = XKO * XL * CTHT                                     RCS3 041
      IF (TERM3 .LE. 1.0E-6) GO TO 10                             RCS3 042
      TERM4 = (SIN(TERM3)) / TERM3                                RCS3 043
      GO TO 20                                                    RCS3 044
   10 TERM4 = 1.0                                                 RCS3 045
C     COMPUTE PHASE TERM FOR CYLINDER(TERM5) AND SPHEROID(TERM8)  RCS3 046
   20 TERM5 = (2.0 * TERM1) + (PI / 4.0)                          RCS3 047
```

```
      IF (TERM3 .LE. 1.0E-6) GO TO 10                                   RCS3 042
      TERM4 = (SIN(TERM3)) / TERM3                                      RCS3 043
      GO TO 20                                                          RCS3 044
   10 TERM4 = 1.0                                                       RCS3 045
C     COMPUTE PHASE TERM FOR CYLINDER(TERM5) AND SPHEROID(TERM8)        RCS3 046
   20 TERM5 = (2.0 * TERM1) + (PI / 4.0)                                RCS3 047
      TERM8 = TERM3 + (2.0 * XKO * SQRT(TERM6))                         RCS3 048
C     COMPUTE REAL AND IMAGINARY PARTS OF CYLINDRICAL (FIRSTR,FIRSTI)   RCS3 049
C              AND UPDATE SPHEROIDAL (SECNDR,SECNDI) RETURNS            RCS3 050
      FIRSTR = TERM2 * TERM4 * COS(TERM5)                               RCS3 051
      FIRSTI = TERM2 * TERM4 * SIN(TERM5)                               RCS3 052
      SECNDR = TERM7 * COS(TERM8)                                       RCS3 053
      SECNDI = TERM7 * SIN(TERM8)                                       RCS3 054
C                                                                       RCS3 055
C     SUM RETURNS AND CONVERT FROM FEET TO METERS                       RCS3 056
      EVVR(I) = (FIRSTR + SECNDR) * 0.304801                            RCS3 057
      EVVI(I) = (FIRSTI + SECNDI) * 0.304801                            RCS3 058
      EHHR(I) = EVVR(I)                                                 RCS3 059
      EHHI(I) = EVVI(I)                                                 RCS3 060
  100 CONTINUE                                                          RCS3 061
C                                                                       RCS3 062
      WRITE (6, 2000) THETA                                             RCS3 063
 2000 FORMAT ( '       ASPECT ANGLE = ', F6.2, ' DEG')                  RCS3 064
C                                                                       RCS3 065
      RETURN                                                            RCS3 066
C                                                                       RCS3 067
```

3

L.2 HEMISPHERE - CYLINDER

The far-field scattering from a hemisphere - cylinder has been formulated using the Ruck-Ufimtsev formulation of the scattering from a cylinder and a modified expression has been utilized in describing the hemispherical returns (Ref 2). The resulting expression of the scattered field is the following:

$$e(\theta)_{\left\{ V \atop H \right\}} = \mp 2\sqrt{\pi} \left\{ \frac{a}{2\sqrt{3}} e^{i2kh\cos\theta} \left[\frac{2}{3} B_{21+} \pm \left(\frac{1}{0.5+\cos\frac{4\theta}{3}} \right) B_{01-} \right. \right.$$

$$\left. + Q_+ \left\{ \frac{2}{3} B_{21-} \pm \left(\frac{1}{0.5+\cos\frac{2(\pi-2\theta)}{3}} \right) B_{01+} \right\} \right]$$

$$\mp \frac{a}{4} \tan\theta \, B_{01-} e^{-i2kh\cos\theta}$$

$$\left. \mp Q_-(S) \pm Q_+ \left({CW_v \atop o} \right) \right\}$$

where $B_{21\pm} = J_2(\zeta) \pm i\, J_1(\zeta)$

$B_{01\pm} = J_o(\zeta) \pm i\, J_1(\zeta)$

$\zeta = 2ka \sin\theta$

$Q_+ = Q(2ka(\theta - \pi/2)$

$Q_- = Q(2ka(\pi/2 - \theta))$

$k = 2\pi/\lambda$ = wave number.

In the primary equation, the upper and lower signs designate the vertically and horizontally polarized returns respectively. The factors S and CW are the sphere specular and spherical creeping-wave returns respectively. They are expanded as follows:

$$S = \frac{a}{2}\left\{1 - \frac{i}{2ka}\right\} e^{-i2k(a+h\cos\theta)}$$

$$CW_v = \frac{a}{2}(2.7162)K\left\{(1+\frac{0.54555}{K^2}) + i\frac{0.94489}{K^2}\right\}e^{-i2kh\cos\theta}\cdot$$

$$\exp\left\{-(2.20075K-\frac{0.44525}{K}) + i(\pi ka+\frac{\pi}{3}+1.27065K+\frac{0.25707}{K})\right\}$$

where $K = (ka)^{1/3}$ and the geometry of the problem is shown in Figure L.2-1. The inputs, outputs, restrictions, and definition ot key terms in the subroutine are presented in the following paragraphs.

L.2.1 INPUTS

The subroutine inputs are read from cards or passed in a common block. The parameters passed in the common block are FC, DF, NMIN, NMAX and are described in Appendix L.1.1.

The inputs read from a card are the following:

	MATHEMATICAL SYMBOL	VARIABLE	DEFINITION	COLUMNS
Card 1	a	A	Radius of hemisphere (inches)	1-10
	h	H	1/2 length of cylinder (inches)	11-20
	θ	ASPECT	Azimuth angle (degrees)	21-30

L.2.2 OUTPUTS

The data base output consists of four linear arrays, EVVR, EVVI, EHHR, EHHI, which contain the real and imaginary parts of the vertically and horizontally polarized backscattered fields (in meters) at frequency increments between NMIN and NMAX.

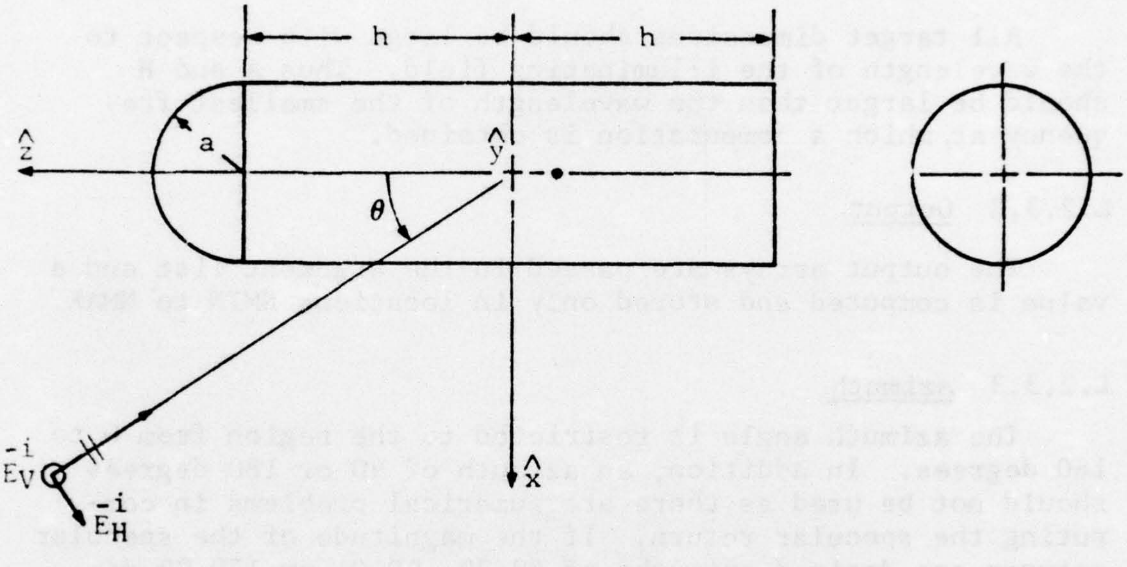


Fig. L.2.1 HEMISPHERE-CYLINDER TARGET GEOMETRY

In addition to the data-base output, if the print option is selected, the frequency, and cross-section (in dBsm) and phase for the case of both vertical and horizontal polarization will be listed.

L.2.3 Restrictions

L.2.3.1 Physical Dimensions

All target dimensions should be large with respect to the wavelength of the illuminating field. Thus A and H should be larger than the wavelength of the smallest frequency at which a computation is obtained.

L.2.3.2 Output

The output arrays are passed in the argument list and a value is computed and stored only in locations NMIN to NMAX.

L.2.3.3 Azimuth

The azimuth angle is restricted to the region from 0 to 180 degrees. In addition, an azimuth of 90 or 180 degrees should not be used as there are numerical problems in computing the specular return. If the magnitude of the specular returns are desired, azimuths of 89.99, 90.01 or 179.99 degrees should be used.

L.2.4 Definition of Key Terms Used in Subroutine

$$\text{TERM 4} = \left(\frac{1}{0.5+\cos\frac{4\theta}{3}}\right)$$

$$\text{TERM 5} = \left(\frac{1}{0.5+\cos\frac{2(\pi-2\theta)}{3}}\right)$$

$$\text{PHASE 1} = \frac{a}{2\sqrt{3}}e^{i2kh\ \cos\theta}$$

$$\text{TERM 2} = \zeta = 2ka\ \sin\theta$$

$$\text{COEFF1} = B_{01+} = J_o(\zeta) + i\,J_1(\zeta)$$

$$\text{FFVV1} = \frac{a}{2\sqrt{3}} e^{i2kh\,\cos\theta} \left[\frac{2}{3} B_{21+} + \left(\frac{1}{0.5+\cos\frac{4\theta}{3}} \right) B_{01-} \right.$$

$$\left. + Q_+ \left\{ \frac{2}{3} B_{21-} + \left(\frac{1}{0.5+\cos\frac{2(\pi-2\theta)}{3}} \right) B_{01+} \right\} \right]$$

$$\text{FFVV2} = -\frac{a}{4} \tan\theta\, B_{01-} e^{-i2kh\,\cos\theta}$$

$$\text{ARGMNT*PHASE 3} = \frac{a}{2} \left\{ 1 - \frac{i}{2ka} \right\} e^{-i2k(a+h\,\cos\theta)}$$

$$\text{PART 1} = \frac{a}{2}(2.7162)K \quad \exp\left\{ -(2.20075K - \frac{0.44525}{K}) \right.$$

$$\text{PHASE 4} = \exp\left\{ +i(\pi ka + \frac{\pi}{3} + 1.27065K + \frac{0.25707}{K}) \right\} \cdot e^{-i2kh\,\cos\theta}$$

$$\text{YY} = \left\{ (1 + \frac{0.54555}{K^2}) + i\frac{0.94489}{K^2} \right\}$$

$$\text{Q PLUS} = Q_+ = Q(2ka(\theta - \pi/2)$$

$$\text{QMINUS} = Q_- = Q(2ka(\pi/2 - \theta))$$

L.2.5 Subroutines Utilized

Subfunctions:

Q(X) computes the exponential smoothing function Q for real argument x

Subroutines:

BESL (TERM 2, XJ0, XJ1, XJ2) returns

J0 (TERM 2) in XJ0

J1 (TERM 2) in XJ0

J1 (TERM 2) in XJ1

J2 (TERM 2) in XJ2

```
      SUBROUTINE TARGEI (EVVR, EVVI, EHHR, EHHI)

      * * TARGET ST-5, HEMISPHERE CYLINDER, UFIMTSEV TECHNIQUE * *

      COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, T0
      NMIN = MINIMUM FREQUENCY SAMPLE
      NMAX = MAXIMUM FREQUENCY SAMPLE
      DF   = FREQUENCY INCREMENT IN MHZ
      FC   = CARRIER FREQUENCY IN GHZ
      DIMENSION EVVR(512), EVVI(512), EHHR(512), EHHI(512)
     1, FREQ(512), SIGMAV(512), SIGMAH(512), PHASEV(512), PHASEH(512)
      COMPLEX COEFF1, COEFF2, PHASE1, PHASE2, PHASE3, PHASE4, FFVV, FFHH
     1, FFVV1, FFHH1,  FFVV2,  FFHH2,  FFVV3,  FFHH3,  FFVV4, CFVV, CFHH
     1 , YY

      A      = RADIUS OF HEMISPHERE
      H      = HALF THE CYLINDER LENGTH
      ASPECT = AZIMUTH ANGLE
      M1     = PRINT OPTION
      ** DIMENSIONS ARE IN INCHES AND ANGLE IS IN DEGREES * *

      READ(5,1000) A,H,ASPECT,M1
 1000 FORMAT( 3F10.0,I5)
      WRITE (6, 1111) A, H, ASPECT, M1
 1111 FORMAT(1H0, ' A = ',  F15.5 , 10X,  ' H = ', F15.5  ,//
     1         1H0, ' THETA = ', F15.5, 15X, ' PRINT OPTION = ', I5

      PI     = 3.14159265358979
      PISQRT = SQRT(PI)
      PIOVR2 = PI / 2.0
      PIOVR3 = PI / 3.0
      S      = -(2.0 / 3.0)
      DTR    = PI / 180.0

      THETA  = ASPECT * DTR
      STHT   = SIN(THETA)
      CTHT   = COS(THETA)
      ZZ      = (A/4.0) * (STHT/CTHT)
      COMPUTE EDGE DIFFRACTION COEFFICIENTS
      THT403 = (4.0 * THETA) / 3.0
      THT3U4 = (2.0 * PIOVR3) - THT403
      TERM1  = (A * SIN(PIOVR3)) / 3.0
      TERM4  = 1.0 / (0.5 + COS(THT403))
      TERM5  = 1.0 / (0.5 + COS(THT3U4))
      TERM6  = S + TERM4
      TERM7  = S - TERM4
      TERM8  = S + TERM5
      TERM9  = S - TERM5
      C      = 11.80285078

      X2KCA   = 2.0 * (0.53234454*FC)       * A
      COMPUTE Q (SMOOTHING) FUNCTIONS
      ZP      = X2KCA * (THETA - PIOVR2)
      ZM      = X2KCA * (PIOVR2 - THETA)
      QPLUS   = Q(ZP)
      QMINUS  = Q(ZM)
```

```
C
C      FREQUENCY LOOP
       DO 900 I = NMIN, NMAX
       XI      = I - 1
       W       = (2.0 * PI * XI * DF) / 1000.0
       FREQ1   = XI * DF / 1000.0
       XK0     = W / C
       P       = (XK0 * A) ** (1.0 / 3.0)
       P2       = P * P
       PART1   = (A / 2.0) * 2.7162 * P * EXP((-2.20075*P) + (0.44525/P))
       TERM0   = 2.0 * XK0 * A
       TERM2   = TERM0 * STHT
       TERM3   = 2.0 * XK0 * H * CTHT
       ARGMNT   = (A/2.0) * SQRT(1.0 + (1.0 /(TERM0*TERM0)))
C
       XJ0     = 0.0
       XJ1     = 0.0
       XJ2     = 0.0
       CALL BESL (TERM2, XJ0, XJ1, XJ2)
C
       COEFF1 = CMPLX(XJ0, XJ1)
       COEFF2 = CONJG(COEFF1)
       COEFF3 = (-5) * (XJ0 + XJ2)
C
       FASE1R = TERM1 * COS(TERM3)
       FASE1I = TERM1 * SIN(TERM3)
       PHASE1 = CMPLX(FASE1R, FASE1I)
C
       FASE2R  = ZZ * COS(TERM3)
       FASE2I  = ZZ * (-SIN(TERM3))
       PHASE2 = CMPLX(FASE2R, FASE2I)
C
       ARG     = TERM0 + TERM3 - ATAN2(-1.0, TERM0)
       FASE3R = COS(ARG)
       FASE3I = -SIN(ARG)
       PHASE3 = CMPLX(FASE3R, FASE3I)
C
       ARG     = PIOVR2+(1.27065*P)+(0.25707/P)+(TERM0*PIOVR2)-TERM3
       FASE4R = COS(ARG)
       FASE4I = +SIN(ARG)
       PHASE4 = CMPLX(FASE4R, FASE4I)
C
C      CYLINDRICAL EDGE + SURFACE RETURN FROM BACK END
       FFVV1  = (QPLUS*(COEFF1*TERM8+COEFF3)+COEFF2*TERM6+COEFF3)*PHASE1
       FFHH1  = (QPLUS*(COEFF1*TERM9+COEFF3)+COEFF2*TERM7+COEFF3)*PHASE1
C
C      CYLINDRICAL SURFACE RETURN (CYLINDER-HEMISPHERE JUNCTION)
       FFVV2  = COEFF2 * PHASE2
       FFHH2  = FFVV2
C
C      HEMISPHERE SPECULAR
       FFVV3    = ARGMNT * PHASE3 * QMINUS
       FFHH3  = FFVV3
C
C      HEMISPHERE CREEPING WAVE
       YY       = CMPLX ( (1.0 +(0.54555/P2)) , (0.94489/P2) )
       FFVV4  = QPLUS * PART1 *    YY  * PHASE4
C
```

```
      FFVV       = 0.025406*(-2.0*PISQRT)*(FFVV1-FFVV2-FFVV3+FFVV4)
      FFHH       = 0.025406*( 2.0*PISQRT)*(FFHH1+FFHH2+FFHH3        )
      CFVV    = CONJG(FFVV)
      CFHH    = CONJG(FFHH)
C
      EVVR(I)     = REAL(CFVV)
      EVVI(I)     = AIMAG(CFVV)
      EHHR(I)     = REAL(CFHH)
      EHHI(I)     = AIMAG(CFHH)
C
      IF (M1) 20, 20, 10
   10 CONTINUE
C
C     DETERMINATION OF CROSS SECTION(IN DBSM) AND PHASE(IN DEGREES)
C                     FOR PRINTOUT OF RESPONSE VERSUS FREQUENCY
      FREQ(I)    = (XI + DF) / 1000.0
      SIGMAV(I)= 10.0*ALOG10(EVVR(I)*EVVR(I) + EVVI(I)*EVVI(I))
      SIGMAH(I)= 10.0*ALOG10(EHHR(I)*EHHR(I) + EHHI(I)*EHHI(I))
      PHASEV(I)= 57.29578 * ATAN2 (EVVI(I), EVVR(I))
      PHASEH(I)= 57.29578 * ATAN2 (EHHI(I), EHHR(I))
C
   20 CONTINUE
C
  900 CONTINUE
C
      IF (M1) 40, 40, 30
   30 CONTINUE
C
C     WRITE (6, 1500)
C1500 FORMAT (1H1)
C     WRITE (6, 2000)(FREQ(I), SIGMAV(I), PHASEV(I), SIGMAH(I),PHASEH(I)
C    1                , I=NMIN,NMAX)
C2000 FORMAT (1H , F10.5, 4F10.2)
   40 CONTINUE
C
      RETURN
      END
      SUBROUTINE BESL ( X, B0, B1, B2 )
C
C    * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS
C    * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS
C    * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )
C
      S = 1.0
      IF (X .LT. 0.0) S = -1.0
      X = ABS (X)
C
      IF ( X .GT. 1.E-6 ) GO TO 5
      B0 = 1.0
      B1 = 0.0
      B2 = 0.0
      X = X * S
      RETURN
C
    5 CONTINUE
C
    1 IF ( X .GE. 3.) GO TO 9
      X1 = X/3
      X1 = X1*X1
      B = 1.+ X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479
     1 +      X1*(-.0039444+ X1*2.1E-4 )))) )
      GO TO 10
C
    9 X2 = 3./X
      F0 =  .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*
     1      (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )
      T0 = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573
     1     +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))
      B = F0*COS(T0)/SQRT(X)
C
   10 B0 = B
C
    2 IF ( X .GE. 3. ) GO TO 19
      X1 = X/3
      X1 = X1*X1
      B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*
     1               (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )
      GO TO 20
C
   19 X2 = 3./X
      F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*
     1      (-.00249511 +X2*(.00113653 -.00020033*X2 )))))
      T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879
     1     +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))
      B = F1*COS(T1)/SQRT(X)
C
   20 B1 = B * S
      X = X * S
      B2= (2./X)*B1 - B0
   50 RETURN
      END
      FUNCTION Q(Z)
C
      IF ( Z.GT. 2.) GO TO 10
      IF ( Z.LT.-2.) GO TO 20
      AZ = ABS(Z)
      P = 1.0/(1.0 + .47047*AZ)
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)
      IF (Z) 2,4,6
    2 Q = (1.0 - Y)/2.
      RETURN
    4 Q = .5
      RETURN
    6 Q = (1.0 + Y)/2.
      RETURN
   10 Q = 1.
      RETURN
   20 Q = 0.
      RETURN
      END
```

CHART TITLE - SUBROUTINE TARGET(FVVR,FVVI,FHHR,FHHI)

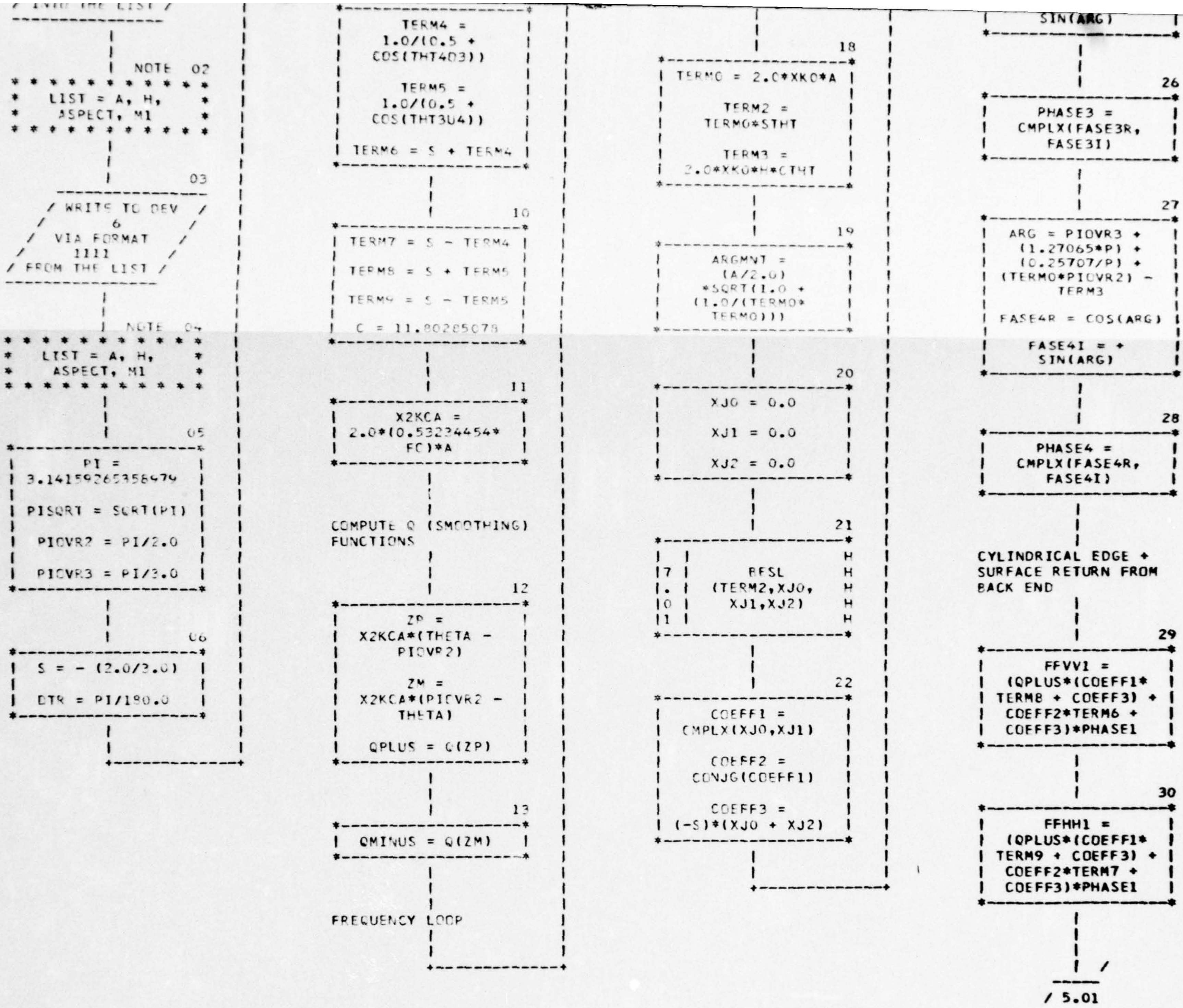
L-16

```
     -----------
    /  TARGET  /
    -----------

          *

 * * TARGET ST-5,
HEMISPHERE CYLINDER,
UFIMTSEV
TECHNIQUE * *


NMIN = MINIMUM
FREQUENCY SAMPLE
NMAX = MAXIMUM
FREQUENCY SAMPLE
DF    = FREQUENCY
INCREMENT IN MHZ
FC    = CARRIER
FREQUENCY IN GHZ


A       = RADIUS OF
HEMISPHERE
H       = HALF THE
CYLINDER LENGTH
ASPECT = AZIMUTH
ANGLE
M1      = PRINT OPTION
 ** DIMENSIONS ARE IN
INCHES AND ANGLE IS
IN DEGREES * *

          *
          |          01
  ------------------
 / READ FROM DEV /
/       5       /
/  VIA FORMAT   /
/     1000      /
/ INTO THE LIST /
------------------
          |
          |  NOTE  02
* * * * * * * * * * * *
*   LIST = A, H,      *
*    ASPECT, M1       *
* * * * * * * * * * * *
          |
```

```
*------------------>*
|                   |              07
|     *--------------------------*
|     |        THETA =           |
|     |      ASPECT*DTR          |
|     |                          |
|     |  STHT = SIN(THETA)       |
|     |                          |
|     |  CTHT = COS(THETA)       |
|     |                          |
|     |         ZZ =             |
|     |       (A/4.0)            |
|     |     *(STHT/CTHT)         |
|     *--------------------------*
|                   |
|                   |
|         COMPUTE EDGE
|         DIFFRACTION
|         COEFFICIENTS
|                   |
|                   |              08
|     *--------------------------*
|     |       THT403 =           |
|     |   (4.0*THETA)/3.0        |
|     |                          |
|     |       THT3U4 =           |
|     |  (2.0*PIOVR3) -          |
|     |        THT403            |
|     |                          |
|     |        TERM1 =           |
|     |  (A*SIN(PIOVR3))         |
|     |          /3.0            |
|     *--------------------------*
|                   |
|                   |              09
|     *--------------------------*
|     |        TERM4 =           |
|     |      1.0/(0.5 +          |
|     |     COS(THT403))         |
|     |                          |
|     |        TERM5 =           |
|     |      1.0/(0.5 +          |
|     |     COS(THT3U4))         |
|     |                          |
|     |  TERM6 = S + TERM4       |
|     *--------------------------*
```

```
*------------------>*
|                   |  NOTE  15
|     * * * * * * * * * * * * *
|     *   BEGIN DO LOOP       *
|     *   900 I = NMIN,       *
|     *        NMAX           *
|     * * * * * * * * * * * * *
|                   |
|       05.13--->|
|                   |              16
|     *--------------------------*
|     |      XI = I - 1          |
|     |                          |
|     |         W =              |
|     |  (2.0*PI*XI*DF)          |
|     |       /1000.0            |
|     |                          |
|     |       FREQ1 =            |
|     |   XI*DF/1000.0           |
|     |                          |
|     |     XKO = W/C            |
|     *--------------------------*
|                   |
|                   |
|                   |              17
|     *--------------------------*
|     |          P =             |
|     |       (XKO*A)            |
|     |     **(1.0/3.0)          |
|     |                          |
|     |     P2 = P*P             |
|     |                          |
|     |       PART1 =            |
|     |       (A/2.0)            |
|     | *2.7162*P*EXP((-         |
|     |   2.20075*P) +           |
|     |   (0.44525/P))           |
|     *--------------------------*
|                   |
|                   |
|                   |              18
|     *--------------------------*
|     | TERMO = 2.0*XKO*A        |
|     |                          |
|     |        TERM2 =           |
|     |     TERMO*STHT           |
|     |                          |
|     |        TERM3 =           |
|     |   2.0*XKO*H*CTHT         |
```

```
*------------------>*
|                   |              23
|     *--------------------------*
|     |        FASE1R =          |
|     | TERM1*COS(TERM3)         |
|     |                          |
|     |        FASE1I =          |
|     | TERM1*SIN(TERM3)         |
|     |                          |
|     |        PHASE1 =          |
|     |   CMPLX(FASE1R,          |
|     |        FASE1I)           |
|     *--------------------------*
|                   |
|                   |              24
|     *--------------------------*
|     |        FASE2R =          |
|     |   ZZ*COS(TERM3)          |
|     |                          |
|     |        FASE2I =          |
|     | ZZ*(-SIN(TERM3))         |
|     |                          |
|     |        PHASE2 =          |
|     |   CMPLX(FASE2R,          |
|     |        FASE2I)           |
|     *--------------------------*
|                   |
|                   |              25
|     *--------------------------*
|     |  ARG = TERMO +           |
|     |        TERM3 -           |
|     | ATAN2(-1.0,TERMO)        |
|     |                          |
|     | FASE3R = COS(ARG)        |
|     |                          |
|     |    FASE3I = -            |
|     |      SIN(ARG)            |
|     *--------------------------*
|                   |
|                   |              26
|     *--------------------------*
|     |        PHASE3 =          |
|     |   CMPLX(FASE3R,          |
|     |        FASE3I)           |
|     *--------------------------*
|                   |
```



```
/ INTO THE LIST /

NOTE 02
LIST = A, H,
ASPECT, M1

03
/ WRITES TO DEV
6
VIA FORMAT
1111
FROM THE LIST /

NOTE 04
LIST = A, H,
ASPECT, M1

05
PI =
3.14159265358979

PISQRT = SQRT(PI)

PIOVR2 = PI/2.0

PIOVR3 = PI/3.0

06
S = - (2.0/3.0)

DTR = PI/180.0
```

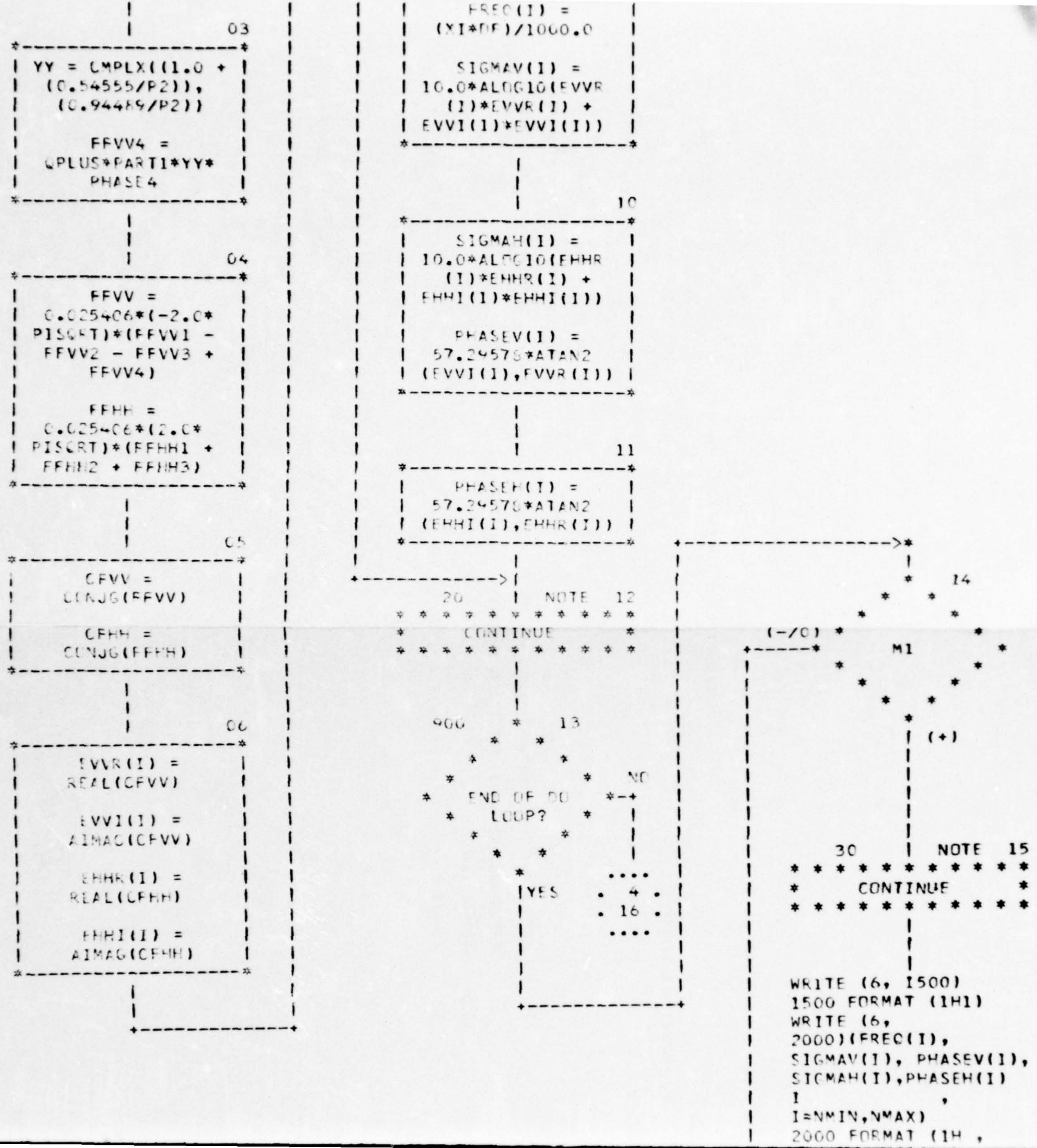
```
TERM4 =
1.0/(0.5 +
COS(THT403))

TERM5 =
1.0/(0.5 +
COS(THT304))

TERM6 = S + TERM4

10
TERM7 = S - TERM4

TERM8 = S + TERM5

TERM9 = S - TERM5

C = 11.80285078

11
X2KCA =
2.0*(0.53234454*
FC)*A

COMPUTE Q (SMOOTHING)
FUNCTIONS

12
ZP =
X2KCA*(THETA -
PIOVR2)

ZM =
X2KCA*(PIOVR2 -
THETA)

QPLUS = Q(ZP)

13
QMINUS = Q(ZM)

FREQUENCY LOOP
```

```
18
TERMO = 2.0*XKO*A

TERM2 =
TERMO*STHT

TERM3 =
2.0*XKO*H*CTHT

19
ARGMNT =
(A/2.0)
*SQRT(1.0 +
(1.0/(TERMO*
TERMO)))

20
XJ0 = 0.0

XJ1 = 0.0

XJ2 = 0.0

21
7.01  BESL
      (TERM2,XJ0,
      XJ1,XJ2)

22
COEFF1 =
CMPLX(XJ0,XJ1)

COEFF2 =
CONJG(COEFF1)

COEFF3 =
(-S)*(XJ0 + XJ2)
```

```
SIN(ARG)

26
PHASE3 =
CMPLX(FASE3R,
FASE3I)

27
ARG = PIOVR3 +
(1.27065*P) +
(0.25707/P) +
(TERMO*PIOVR2) -
TERM3

FASE4R = COS(ARG)

FASE4I = +
SIN(ARG)

28
PHASE4 =
CMPLX(FASE4R,
FASE4I)

CYLINDRICAL EDGE +
SURFACE RETURN FROM
BACK END

29
FFVV1 =
(QPLUS*(COEFF1*
TERM8 + COEFF3) +
COEFF2*TERM6 +
COEFF3)*PHASE1

30
FFHH1 =
(QPLUS*(COEFF1*
TERM9 + COEFF3) +
COEFF2*TERM7 +
COEFF3)*PHASE1

/ 5.01
```

```
CHART TITLE - SUBROUTINE  TARGET(EVVR,EVVI,EHHR,EHHI)




 04.30--->*                 +------------------->*
          |                 |                    |
CYLINDRICAL SURFACE         |                    *   07
RETURN                      |                 *     *
(CYLINDER-HEMISPHERE        |              *           *
JUNCTION)                   |      (-/0) *                *
          |                 |   +------*        M1           *
          |                 |   |        *                *
          |            01   |   |          *           *
*--------------------*      |   |             *     *
|      FFVV2 =       |      |   |                *
|   COEFF2*PHASE2    |      |   |                | (+)
|                    |      |   |                |
|    FFHH2 = FFVV2   |      |   |                |
*--------------------*      |   |                |
          |                 |   |                |
          |                 |   |                |
          |                 |   |         10     |  NOTE  08
HEMISPHERE SPECULAR         |   |   * * * * * * * * * * *
          |                 |   |   *        CONTINUE     *
          |                 |   |   * * * * * * * * * * *
          |            02   |   |                |
*--------------------*      |   |                |
|      FFVV3 =       |      |   |                |
|  ARGMNT*PHASE3*    |      |   |   DETERMINATION OF
|      QMINUS        |      |   |   CROSS SECTION(IN
|                    |      |   |   DBSM) AND PHASE(IN
|    FFHH3 = FFVV3   |      |   |   DEGREES)
*--------------------*      |   |   FOR PRINTOUT OF
          |                 |   |   RESPONSE VERSUS
          |                 |   |   FREQUENCY
          |                 |   |                |
HEMISPHERE CREEPING         |   |                |
WAVE                        |   |                |
          |                 |   |                |        09
          |                 |   |   *--------------------*
          |            03   |   |   |       FREQ(I) =    |
*--------------------*      |   |   |  (XI*DF)/1000.0    |
| YY = CMPLX((1.0 +  |      |   |   |                    |
                            |   |   |      SIGMAV(I) =   |
                            |   |   |  10.0*ALOG10(EVVR
```

2


03
YY = CMPLX((1.0 + (0.54555/P2)), (0.94489/P2))
FFVV4 = QPLUS*PART1*YY* PHASE4
04
FFVV = 0.025406*(-2.0* PISQRT)*(FFVV1 - FFVV2 - FFVV3 + FFVV4)
FFHH = 0.025406*(2.0* PISQRT)*(FFHH1 + FFHH2 + FFHH3)
05
CFVV = CONJG(FFVV)
CFHH = CONJG(FFHH)
06
EVVR(I) = REAL(CFVV)
EVVI(I) = AIMAG(CFVV)
EHHR(I) = REAL(CFHH)
EHHI(I) = AIMAG(CFHH)
FREQ(I) = (XI*DF)/1000.0
SIGMAV(I) = 10.0*ALOG10(EVVR (I)*EVVR(I) + EVVI(I)*EVVI(I))
10
SIGMAH(I) = 10.0*ALOG10(EHHR (I)*EHHR(I) + EHHI(I)*EHHI(I))
PHASEV(I) = 57.29578*ATAN2 (EVVI(I),EVVR(I))
11
PHASEH(I) = 57.29578*ATAN2 (EHHI(I),EHHR(I))
20
NOTE 12
CONTINUE
900
13
END OF DO LOOP?
NO
YES
4
16
14
M1
(-/0)
(+)
30
NOTE 15
CONTINUE
WRITE (6, 1500)
1500 FORMAT (1H1)
WRITE (6,
2000)(FREQ(I),
SIGMAV(I), PHASEV(I),
SIGMAH(I),PHASEH(I)
I ,
I=NMIN,NMAX)
2000 FORMAT (1H ,

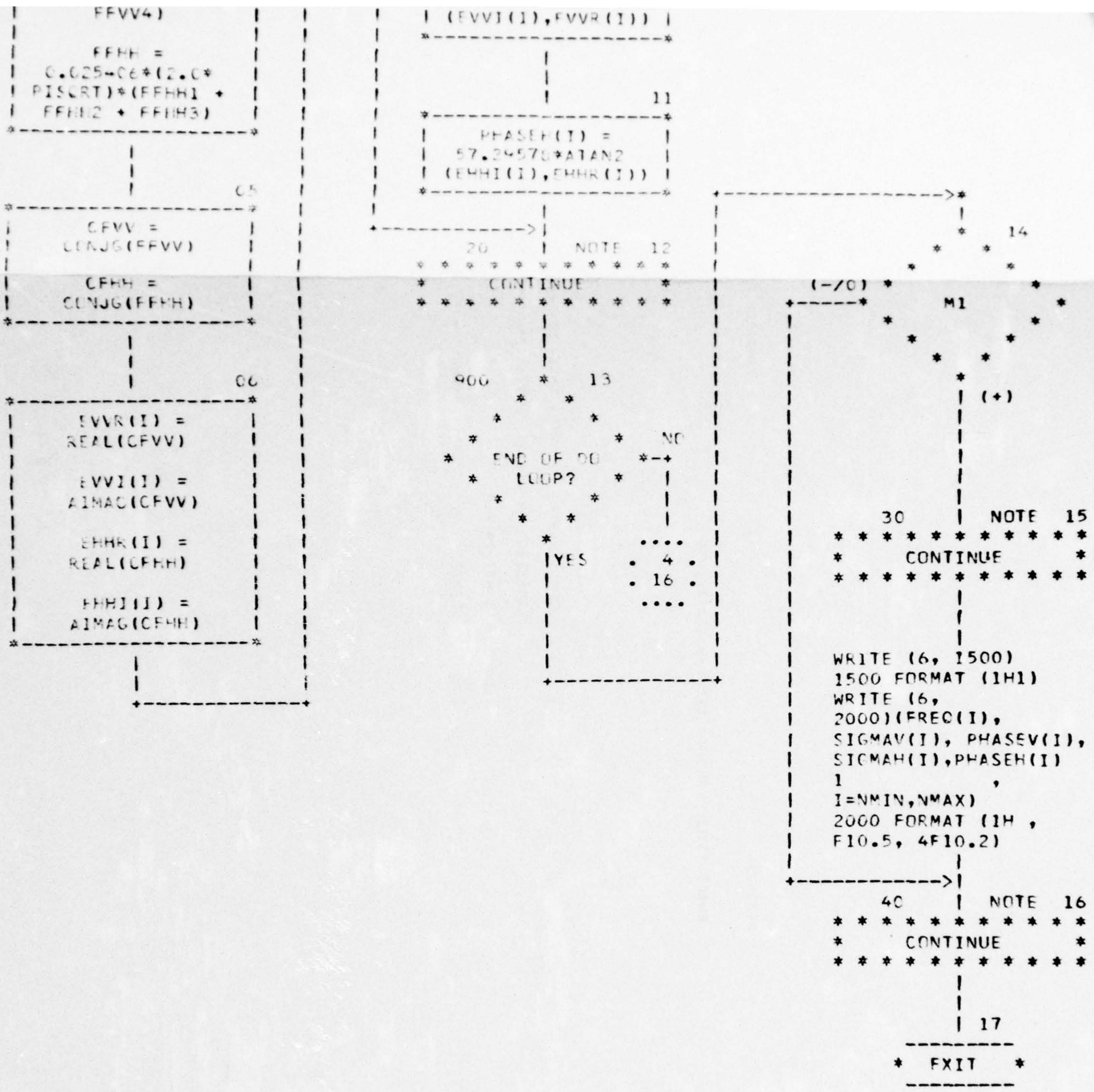
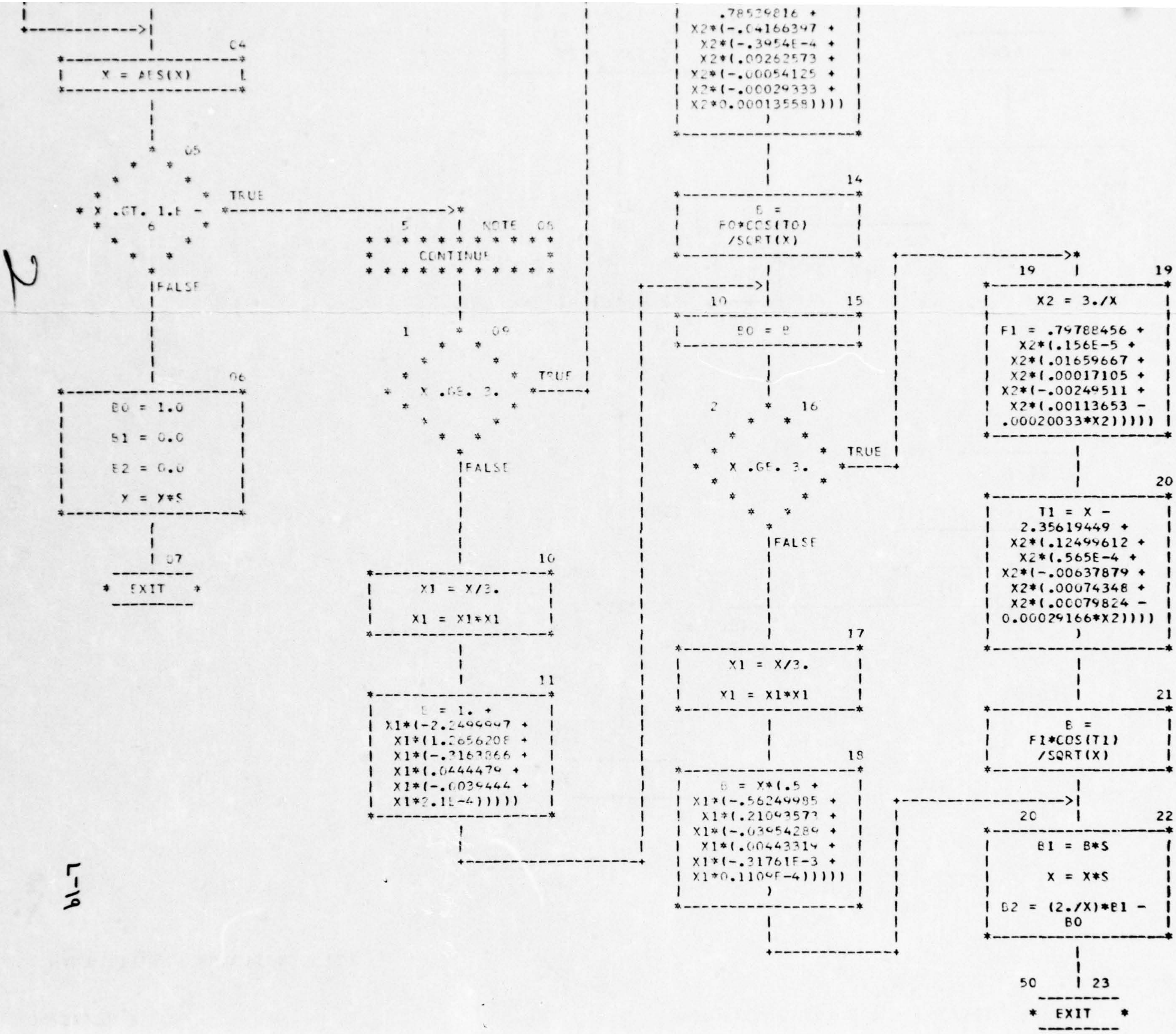

FFVV4)
FFHH =
0.025-06*(2.0*
PISCRT)*(FFHH1 +
FFHH2 + FFHH3)
05
CFVV =
CONJG(FFVV)
CFHH =
CONJG(FFHH)
06
EVVR(I) =
REAL(CFVV)
EVVI(I) =
AIMAG(CFVV)
EHHR(I) =
REAL(CFHH)
EHHI(I) =
AIMAG(CFHH)
(EVVI(I),EVVR(I))
11
PHASEH(I) =
57.29578*ATAN2
(EHHI(I),EHHR(I))
20
NOTE 12
CONTINUE
900
13
END OF DO
LOOP?
NO
YES
4
16
14
(-/0)
M1
(+)
30
NOTE 15
CONTINUE
WRITE (6, 1500)
1500 FORMAT (1H1)
WRITE (6,
2000)(FREQ(I),
SIGMAV(I), PHASEV(I),
SIGMAH(I),PHASEH(I)
1 ,
I=NMIN,NMAX)
2000 FORMAT (1H ,
F10.5, 4F10.2)
40
NOTE 16
CONTINUE
17
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
            COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, TO
            DIMENSION EVVR(512), EVVI(512), EHHR(512), EHHI(512)
1000        FORMAT (4F6.2)
2000        FORMAT ( '       ASPECT ANGLE = ', F6.2, ' DEG')
```

CHART TITLE - SUBROUTINE BESL(X,B0,B1,B2)

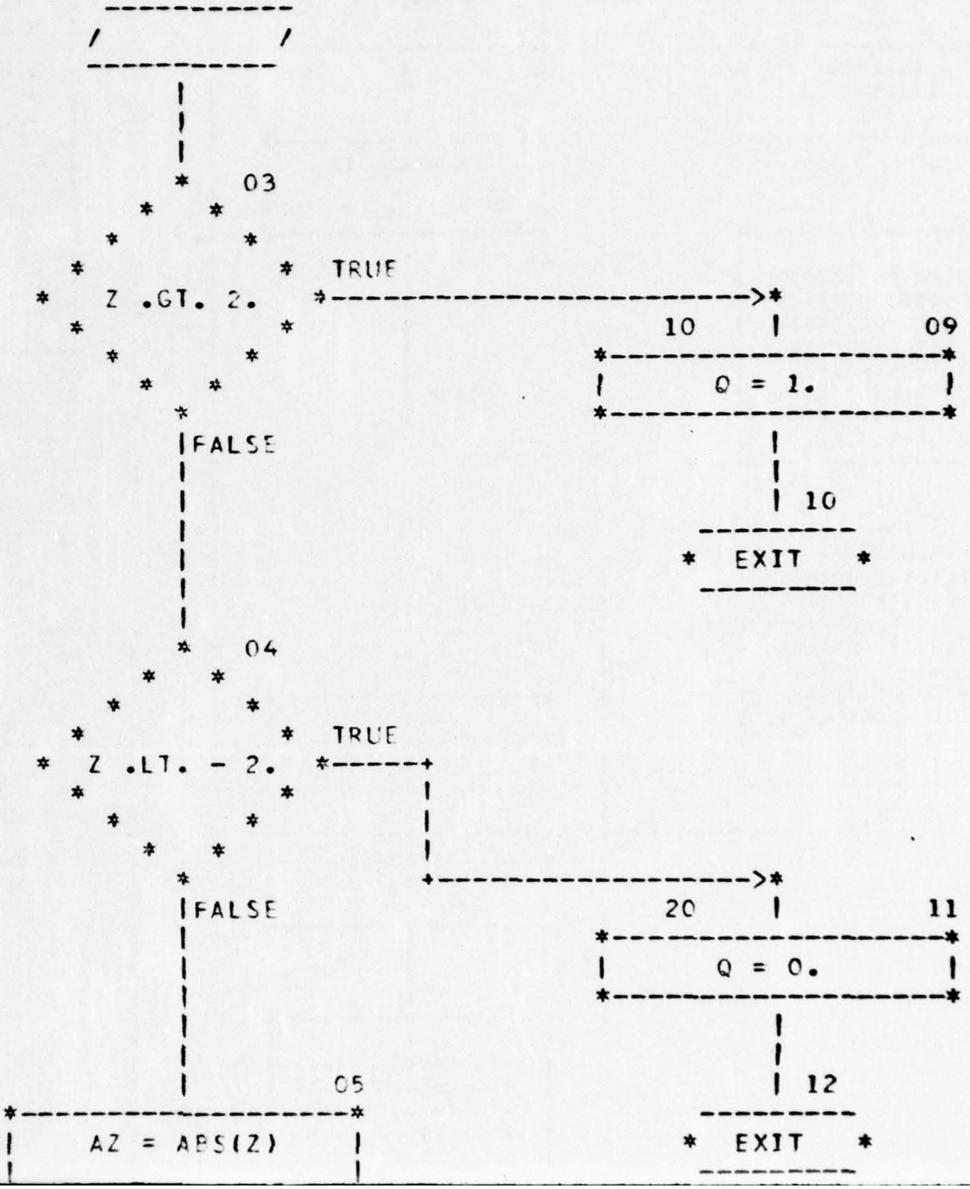
```
        -----------
       /   BESL   /
       -----------

    04.21*-->*

     * BESSEL FUNCTION
    SUBROUTINE UTILIZING
    POLYNOMIAL
    APPROXIMATIONS
     * COMPUTES J0,J1,OR
    J2 FOR POSITIVE REAL
    ARGUMENTS
     * REFERENCE (HNDBK
    MATH FUNCT BY
    ABRAMOWITZ AND STEGUN
    SECTION 9.4 )

              *
              |        01
    *-------------------*
    |     S = 1.0       |
    *-------------------*
              |
              |
              |
              *   02                                +------------------>*
           *     *                                  |          o        |          12
        *           *                               |          *----------------------*
FALSE *              *                              |          |    X2 = 3./X         |
+-----*  X .LT. 0.0    *                            |          |                      |
|      *              *                             |          | FC = .79788456 +     |
|        *          *                               |          |   X2*(-.77E-6 +      |
|          *      *                                 |          | X2*(-.00552740 +     |
|             *                                     |          |  X2*(-.9512E-4 +     |
|             |TRUE                                 |          |  X2*(.00137237 +     |
|             |                                     |          | X2*(-.72805E-3 +     |
|             |                                     |          | X2*0.14476E-3))))    |
|             |                                     |          |        )             |
|             |                                     |          *----------------------*
|             |                                     |                     |
|             |     03                              |                     |
|   *-------------------*                           |                     |          13
|   |    S = - 1.0      |                           |          *----------------------*
|   *-------------------*                           |          |     TC = X -         |
|             |                                     |          |     .78539816 +      |
+------------>|                                     |          | X2*(-.04166397 +     |
              |     04                              |          |  X2*(-.3954E-4 +     |
    *-------------------*                           |          |  X2*(.00262573 +     |
    |    X = ABS(X)     |                           |          | X2*(-.00054125 +     |
    *-------------------*                           |          | X2*(-.00029333 +     |
              |                                     |          | X2*0.00013558))))    |
                                                    |          |        )             |
```


04
X = ABS(X)
05
X .GT. 1.E - 6
TRUE
FALSE
06
B0 = 1.0
B1 = 0.0
B2 = 0.0
Y = Y*S
07
EXIT
5
NOTE 08
CONTINUE
1
09
X .GE. 3.
TRUE
FALSE
10
X1 = X/3.
X1 = X1*X1
11
B = 1. +
X1*(-2.2499997 +
X1*(1.2656208 +
X1*(-.3163866 +
X1*(.0444479 +
X1*(-.0039444 +
X1*2.1E-4)))))
.78539816 +
X2*(-.04166397 +
X2*(-.3954E-4 +
X2*(.00262573 +
X2*(-.00054125 +
X2*(-.00029333 +
X2*0.00013558))))
)
14
B =
F0*COS(T0)
/SQRT(X)
10
15
B0 = B
2
16
X .GE. 3.
TRUE
FALSE
17
X1 = X/3.
X1 = X1*X1
18
B = X*(.5 +
X1*(-.56249985 +
X1*(.21093573 +
X1*(-.03954289 +
X1*(.00443319 +
X1*(-.31761E-3 +
X1*0.1109E-4)))))
)
19
19
X2 = 3./X
F1 = .79788456 +
X2*(.156E-5 +
X2*(.01659667 +
X2*(.00017105 +
X2*(-.00249511 +
X2*(.00113653 -
.00020033*X2)))))
20
T1 = X -
2.35619449 +
X2*(.12499612 +
X2*(.565E-4 +
X2*(-.00637879 +
X2*(.00074348 +
X2*(.00079824 -
0.00029166*X2))))
)
21
B =
F1*COS(T1)
/SQRT(X)
20
22
B1 = B*S
X = X*S
B2 = (2./X)*B1 -
B0
50
23
EXIT

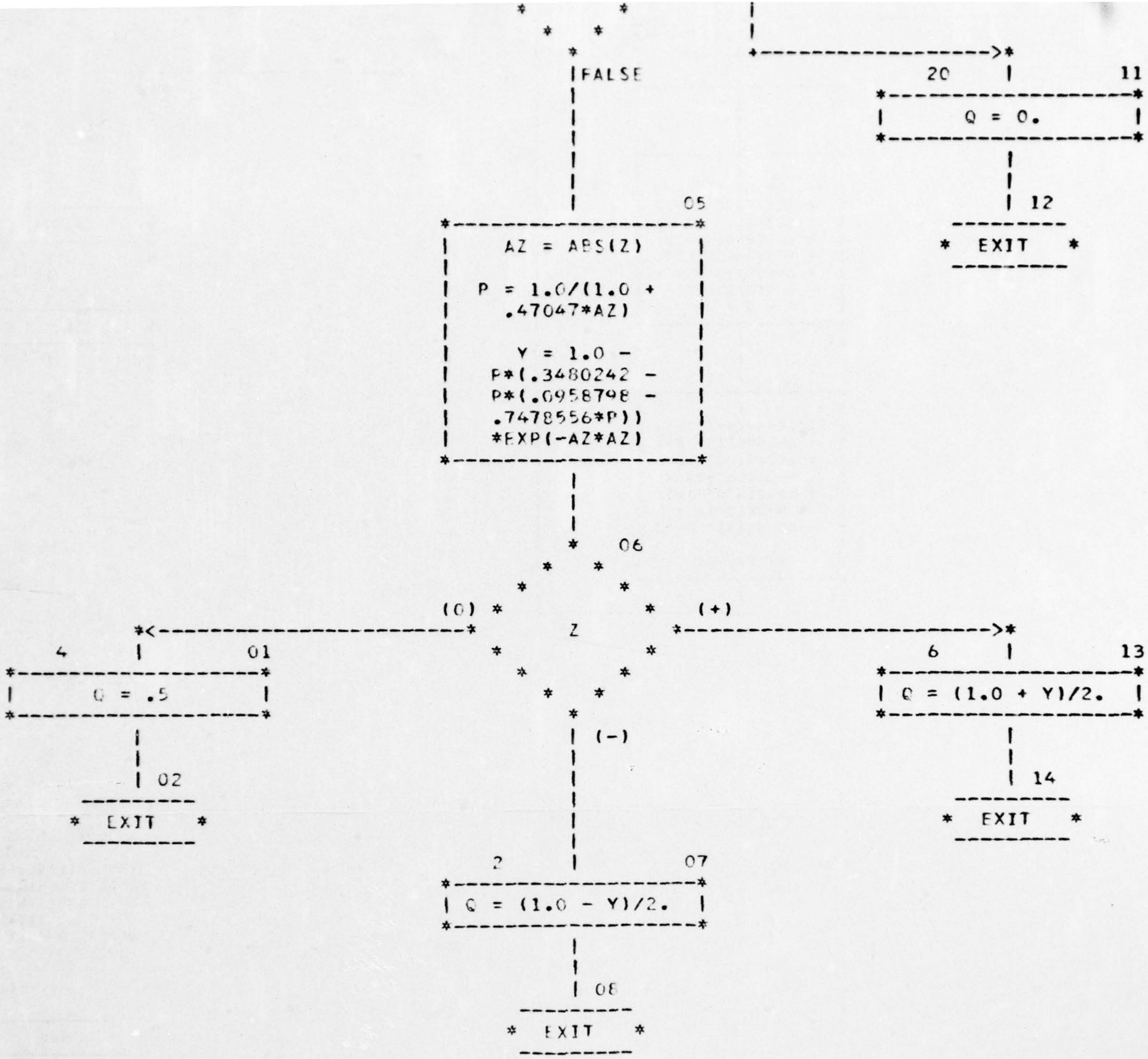
2

CHART TITLE - FUNCTION Q(Z)

L-20



2


FALSE
05
AZ = ABS(Z)
P = 1.0/(1.0 +
.47047*AZ)
Y = 1.0 -
P*(.3480242 -
P*(.0958798 -
.7478556*P))
*EXP(-AZ*AZ)
20
11
Q = 0.
12
EXIT
06
Z
(0)
(+)
(-)
4
01
Q = .5
02
EXIT
6
13
Q = (1.0 + Y)/2.
14
EXIT
2
07
Q = (1.0 - Y)/2.
08
EXIT

```
      SUBROUTINE TARGET (EVVR, EVVI, EHHR, EHHI)                        RCS4 001
C                                                                       RCS4 002
C     * * TARGET ST-5, HEMISPHERE CYLINDER, UFIMTSEV TECHNIQUE * *      RCS4 003
C                                                                       RCS4 004
      COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, TO                       RCS4 005
C     NMIN = MINIMUM FREQUENCY SAMPLE                                   RCS4 006
C     NMAX = MAXIMUM FREQUENCY SAMPLE                                   RCS4 007
C     DF   = FREQUENCY INCREMENT IN MHZ                                 RCS4 008
C     FC   = CARRIER FREQUENCY IN GHZ                                   RCS4 009
      DIMENSION EVVR(512), EVVI(512), EHHR(512), EHHI(512)              RCS4 010
     1, FREQ(512), SIGMAV(512), SIGMAH(512), PHASEV(512), PHASEH(512)   RCS4 011
      COMPLEX COEFF1, COEFF2, PHASE1, PHASE2, PHASE3, PHASE4, FFVV, FFHHRCS4 012
     1, FFVV1, FFHH1,  FFVV2,  FFHH2,  FFVV3,  FFHH3,  FFVV4, CFVV, CFHHRCS4 013
     1 , YY                                                             RCS4 014
C                                                                       RCS4 015
C     A      = RADIUS OF HEMISPHERE                                     RCS4 016
C     H      = HALF THE CYLINDER LENGTH                                 RCS4 017
C     ASPECT = AZIMUTH ANGLE                                            RCS4 018
C     M1     = PRINT OPTION                                             RCS4 019
C     ** DIMENSIONS ARE IN INCHES AND ANGLE IS IN DEGREES * *           RCS4 020
C                                                                       RCS4 021
      READ(5,1000) A,H,ASPECT,M1                                        RCS4 022
 1000 FORMAT( 3F10.0,I5)                                                RCS4 024
      WRITE (6, 1111) A, H, ASPECT, M1                                  RCS4 025
      C                                                                        RCS4 021
            READ(5,1000) A,H,ASPECT,M1                                         RCS4 022
       1000 FORMAT( 3F10.0,I5)                                                 RCS4 024
            WRITE (6, 1111) A, H, ASPECT, M1                                   RCS4 025
       1111 FORMAT(1H0, ' A = ',  F15.5 , 10X,  ' H = ', F15.5   //            RCS4 026
           1        1H0, ' THETA = ', F15.5, 15X, ' PRINT OPTION = ', I5   )   RCS4 027
      C                                                                        RCS4 028
            PI     = 3.14159265358979                                          RCS4 029
            PISQRT = SQRT(PI)                                                  RCS4 030
            PIOVR2 = PI / 2.0                                                  RCS4 031
            PIOVR3 = PI / 3.0                                                  RCS4 032
            S      = -(2.0 / 3.0)                                              RCS4 033
            DTR    = PI / 180.0                                                RCS4 034
      C                                                                        RCS4 035
            THETA  = ASPECT * DTR                                              RCS4 036
            STHT   = SIN(THETA)                                                RCS4 037
            CTHT   = COS(THETA)                                                RCS4 038
            ZZ      = (A/4.0) * (STHT/CTHT)                                    RCS4 039
      C     COMPUTE EDGE DIFFRACTION COEFFICIENTS                              RCS4 040
            THT4O3 = (4.0 * THETA) / 3.0                                       RCS4 041
            THT3O4 = (2.0 * PIOVR3) - THT4O3                                   RCS4 042
            TERM1  = (A * SIN(PIOVR3)) / 3.0                                   RCS4 043
            TERM4  = 1.0 / (0.5 + COS(THT4O3))                                 RCS4 044
            TERM5  = 1.0 / (0.5 + COS(THT3O4))                                 RCS4 045
            TERM6  = S + TERM4                                                 RCS4 046
            TERM7  = S - TERM4                                                 RCS4 047
            TERM8  = S + TERM5                                                 RCS4 048
```

2

L-22

```
CARD NO      ****                        CONTENTS                               ****

  116              TERM9   = S - TERM5                                         RCS4 049
  117              C       = 11.80285078                                       RCS4 050
  118          C                                                               RCS4 051
  119              X2KCA    = 2.0 * (0.53234454*FC)       * A                  RCS4 052
  120          C   COMPUTE Q (SMOOTHING) FUNCTIONS                             RCS4 053
  121              ZP       = X2KCA * (THETA - PIOVR2)                         RCS4 054
  122              ZM       = X2KCA * (PIOVR2 - THETA)                         RCS4 055
  123              QPLUS    = Q(ZP)                                            RCS4 056
  124              QMINUS   = Q(ZM)                                            RCS4 057
  125          C                                                               RCS4 058
  126          C   FREQUENCY LOOP                                              RCS4 059
  127              DO 900 I = NMIN, NMAX                                       RCS4 060
  128              XI      = I - 1                                             RCS4 061
  129              W       = (2.0 * PI * XI * DF) / 1000.0                     RCS4 062
  130              FREQ1   = XI * DF / 1000.0                                  RCS4 063
  131              XKC     = W / C                                             RCS4 064
  132              P       = (XKO * A) ** (1.0 / 3.0)                          RCS4 065
  133              P2       = P * P                                            RCS4 066
  134              PART1   = (A / 2.0) * 2.7162 * P * EXP((-2.20075*P) + (0.44525/P)) RCS4 067
  135              TERMO   = 2.0 * XKO * A                                     RCS4 068
  136              TERM2   = TERMO * STHT                                      RCS4 069
  137              TERM3   = 2.0 * XKO * H * CTHT                              RCS4 070
```

```
      TERM3  = 2.0 * XKC * H * CTHT                                    RCS4 070
      ARGMNT  = (A/2.0) * SQRT(1.0 + (1.0 /(TERMO*TERMO)))              RCS4 071
C                                                                      RCS4 072
      XJO    = 0.0                                                     RCS4 073
      XJ1    = 0.0                                                     RCS4 074
      XJ2    = 0.0                                                     RCS4 075
      CALL BESL (TERM2, XJO, XJ1, XJ2)                                 RCS4 076
C                                                                      RCS4 077
      COEFF1 = CMPLX(XJO, XJ1)                                         RCS4 078
      COEFF2 = CONJG(COEFF1)                                           RCS4 079
      COEFF3 = (-S) * (XJO + XJ2)                                      RCS4 080
C                                                                      RCS4 081
      FASE1R = TERM1 * COS(TERM3)                                      RCS4 082
      FASE1I = TERM1 * SIN(TERM3)                                      RCS4 083
      PHASE1 = CMPLX(FASE1R, FASE1I)                                   RCS4 084
C                                                                      RCS4 085
      FASE2R  = ZZ * COS(TERM3)                                        RCS4 086
      FASE2I  = ZZ * (-SIN(TERM3))                                     RCS4 087
      PHASE2 = CMPLX(FASE2R, FASE2I)                                   RCS4 088
C                                                                      RCS4 089
      ARC    = TERMC + TERM3 - ATAN2(-1.0, TERMO)                      RCS4 090
      FASE3R = COS(ARG)                                                RCS4 091
      FASE3I = -SIN(ARG)                                               RCS4 092
      PHASE3 = CMPLX(FASE3R, FASE3I)                                   RCS4 093
C                                                                      RCS4 094
      ARC    = PIOVR3+(1.27065*P)+(0.25707/P)+(TERMO*PIOVR2)-TERM3    RCS4 095
      FASE4R = COS(ARG)                                                RCS4 096
      FASE4I = +SIN(ARG)                                               RCS4 097
      FASE2I  = Z2 * (-SIN(TERM3))                                      RCS4 087
      PHASE2 = CMPLX(FASE2R, FASE2I)                                    RCS4 088
C                                                                       RCS4 089
      ARC     = TERMC + TERM3 - ATAN2(-1.0, TERMO)                      RCS4 090
      FASE3R = COS(ARG)                                                 RCS4 091
      FASE3I = -SIN(ARG)                                                RCS4 092
      PHASE3 = CMPLX(FASE3R, FASE3I)                                    RCS4 093
C                                                                       RCS4 094
      ARC     = PIOVR3+(1.27065*P)+(0.25707/P)+(TERMO*PIOVR2)-TERM3     RCS4 095
      FASE4R = COS(ARG)                                                 RCS4 096
      FASE4I = +SIN(ARG)                                                RCS4 097
      PHASE4 = CMPLX(FASE4R, FASE4I)                                    RCS4 098
C                                                                       RCS4 099
C     CYLINDRICAL EDGE + SURFACE RETURN FROM BACK END                   RCS4 100
      FFVV1   = (QPLUS*(COEFF1*TERM8+COEFF3)+COEFF2*TERM6+COEFF3)*PHASE1 RCS4 101
      FFHH1   = (QPLUS*(COEFF1*TERM9+COEFF3)+COEFF2*TERM7+COEFF3)*PHASE1 RCS4 102
C                                                                       RCS4 103
C     CYLINDRICAL SURFACE RETURN (CYLINDER-HEMISPHERE JUNCTION)         RCS4 104
      FFVV2   = COEFF2 * PHASE2                                         RCS4 105
      FFHH2   = FFVV2                                                   RCS4 106
```

3

```
CARD NO      ****                      CONTENTS                              ****

             C                                                           RCS4 107
             C      HEMISPHERE SPECULAR                                  RCS4 108
                    FFVV3   = ARGMNT * PHASE3 * CMINUS                   RCS4 109
                    FFHH3  = FFVV3                                       RCS4 110
             C                                                           RCS4 111
             C      HEMISPHERE CREEPING WAVE                             RCS4 112
                    YY      = CMPLX ( (1.0 +(0.54555/P2)) , (0.94459/P2) )   RCS4 113
                    FFVV4  = CPLUS * PART1 *   YY  * PHASE4              RCS4 114
             C                                                           RCS4 115
                    FFVV    = 0.025406*(-2.0*PISQRT)*(FFVV1-FFVV2-FFVV3+FFVV4)   RCS4 116
                    FFHH    = 0.025406*( 2.0*PISQRT)*(FFHH1+FFHH2+FFHH3      )   RCS4 117
                    CFVV   = CONJG(FFVV)                                 RCS4 118
                    CFHH   = CONJG(FFHH)                                 RCS4 119
             C                                                           RCS4 120
                    EVVR(I)   = REAL(CFVV)                               RCS4 121
                    EVVI(I)   = AIMAG(CFVV)                              RCS4 122
                    FHHR(I)   = REAL(CFHH)                               RCS4 123
                    FHHI(I)   = AIMAG(CFHH)                              RCS4 124
             C                                                           RCS4 125
                    IF (M1) 20, 20, 10                                   RCS4 126
                 10 CONTINUE                                             RCS4 127
```

```
C                                                                       RCS4 125
      IF (MI) 20, 20, 10                                                RCS4 126
   10 CONTINUE                                                          RCS4 127
C                                                                       RCS4 128
C     DETERMINATION OF CROSS SECTION(IN DBSM) AND PHASE(IN DEGREES)     RCS4 129
C                         FOR PRINTOUT OF RESPONSE VERSUS FRECUENCY     RCS4 130
      FREQ(I)    = (XI * DF) / 1000.0                                   RCS4 131
      SIGMAV(I)= 10.0*ALOG10(EVVR(I)*EVVR(I) + EVVI(I)*EVVI(I))         RCS4 132
      SIGMAH(I)= 10.0*ALOG10(EHHR(I)*EHHR(I) + EHHI(I)*EHHI(I))         RCS4 133
      PHASEV(I)= 57.29578 * ATAN2 (EVVI(I), EVVR(I))                    RCS4 134
      PHASEH(I)= 57.29578 * ATAN2 (EHHI(I), EHHR(I))                    RCS4 135
C                                                                       RCS4 136
   20 CONTINUE                                                          RCS4 137
C                                                                       RCS4 138
  900 CONTINUE                                                          RCS4 139
C                                                                       RCS4 140
      IF (MI) 40, 40, 30                                                RCS4 141
   30 CONTINUE                                                          RCS4 142
C                                                                       RCS4 143
C     WRITE (6, 1500)                                                   RCS4 144
C1500 FORMAT (1H1)                                                      RCS4 145
C     WRITE (6, 2000)(FREQ(I), SIGMAV(I), PHASEV(I), SIGMAH(I),PHASEH(I)RCS4 146
C    1            , I=NMIN,NMAX)                                        RCS4 147
C2000 FORMAT (1H , F10.5, 4F10.2)                                       RCS4 148
   40 CONTINUE                                                          RCS4 149
C                                                                       RCS4 150
      RETURN                                                            RCS4 151
      END                                                               RCS4 152
```

2

```
04/26/76          INPUT LISTING                          AUTOFLOW CHART SET - FWD/SCL  RADSIM

CARD NO      ****                          CONTENTS                                     ****
  220                SUBROUTINE BESL ( X, B0, B1, B2 )                              RCS4 153
  221          C                                                                    RCS4 154
  222          C   * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS  RCS4 155
  223          C   * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS               RCS4 156
  224          C   * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )RCS4 157
  225          C                                                                    RCS4 158
  226                S = 1.0                                                        RCS4 159
  227                IF (X .LT. 0.0) S = -1.0                                       RCS4 160
  228                X = ABS (X)                                                    RCS4 161
  229          C                                                                    RCS4 162
  230                IF ( X .GT. 1.E-6 ) GO TO 5                                    RCS4 163
  231                B0 = 1.0                                                       RCS4 164
  232                B1 = 0.0                                                       RCS4 165
  233                B2 = 0.0                                                       RCS4 166
  234                X = X * S                                                      RCS4 167
  235                RETURN                                                         RCS4 168
  236          C                                                                    RCS4 169
  237              5 CONTINUE                                                       RCS4 170
  238          C                                                                    RCS4 171
  239              1 IF ( X .GE. 3.) GO TO 9                                        RCS4 172
  240                X1 = X/3.                                                      RCS4 173
  241                X1 = X1*X1                                                     RCS4 174
  242                B = 1.+ X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479RCS4 175
    1 IF ( X .GT. 3.) GO TO 9                                           RCS4 172
      X1 = X/3.                                                         RCS4 173
      X1 = X1*X1                                                        RCS4 174
      B = 1.+ X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479RCS4 175
     1 +       X1*(-.0039444+ X1*2.1E-4 )))) )                          RCS4 176
      GO TO 10                                                          RCS4 177
C                                                                       RCS4 178
    9 X2 = 3./X                                                         RCS4 179
      F0 =  .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*  RCS4 180
     1       (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )           RCS4 181
      T0 = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573  RCS4 182
     1    +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))          RCS4 183
      B = F0*COS(T0)/SQRT(X)                                            RCS4 184
C                                                                       RCS4 185
   10 B0 = B                                                            RCS4 186
C                                                                       RCS4 187
    2 IF ( X .GE. 3. ) GO TO 19                                         RCS4 188
      X1 = X/3.                                                         RCS4 189
      X1 = X1*X1                                                        RCS4 190
      B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*    RCS4 191
     1                  (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) ) RCS4 192
      GO TO 20                                                          RCS4 193
C                                                                       RCS4 194
   19 X2 = 3./X                                                         RCS4 195
      F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*    RCS4 196
     1       (-.00249511 +X2*(.00113653 -.00020033*X2 )))))             RCS4 197
      T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879   RCS4 198
     1    +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))            RCS4 199
     1    (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )          RCS4 181
      T0 = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573 RCS4 182
     1    +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))        RCS4 183
      B = F0*COS(T0)/SQRT(X)                                          RCS4 184
C                                                                     RCS4 185
   10 B0 = B                                                          RCS4 186
C                                                                     RCS4 187
    2 IF ( X .GE. 3. ) GO TO 19                                       RCS4 188
      X1 = X/3.                                                       RCS4 189
      X1 = X1*X1                                                      RCS4 190
      B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*  RCS4 191
     1                   (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) ) RCS4 192
      GO TO 20                                                        RCS4 193
C                                                                     RCS4 194
   19 X2 = 3./X                                                       RCS4 195
      F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*  RCS4 196
     1    (-.00249511 +X2*(.00113653 -.00020033*X2 )))))             RCS4 197
      T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879 RCS4 198
     1    +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))          RCS4 199
      B = F1*COS(T1)/SQRT(X)                                          RCS4 200
C                                                                     RCS4 201
   20 B1 = B * S                                                      RCS4 202
      X = X * S                                                       RCS4 203
      B2= (2./X)*B1 - B0                                              RCS4 204
   30 RETURN                                                          RCS4 205
      END                                                             RCS4 206
```

L-23a

3

```
      FUNCTION G(Z)                                                     RCS4 207
C                                                                       RCS4 208
      IF ( Z.GT. 2.) GO TO 10                                           RCS4 209
      IF ( Z.LT.-2.) GO TO 20                                           RCS4 210
      AZ = ABS(Z)                                                       RCS4 211
      P = 1.0/(1.0 + .47047*AZ)                                         RCS4 212
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)    RCS4 213
      IF (Z) 2,4,6                                                      RCS4 214
    2 G = (1.0 - Y)/2.                                                  RCS4 215
      RETURN                                                            RCS4 216
    4 G = .5                                                            RCS4 217
      RETURN                                                            RCS4 218
    6 G = (1.0 + Y)/2.                                                  RCS4 219
      RETURN                                                            RCS4 220
   10 G = 1.                                                            RCS4 221
      RETURN                                                            RCS4 222
   20 G = 0.                                                            RCS4 223
      RETURN                                                            RCS4 224
      END                                                               RCS4 225
```

04/26/76   AUTOFLOW CHART SET - FWO/SCL RADSIM

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
                COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, TC
                DIMENSION EVVR(512), EVVI(512), EHHR(512), EHHI(512)
                , FREQ(512), SIGMAV(512), SIGMAH(512), PHASEV(512), PHASEH(512)
                COMPLEX COEFF1, COEFF2, PHASE1, PHASE2, PHASE3, PHASE4, FFVV, FFHH
                , FFVV1, FFHH1,  FFVV2,  FFHH2,  FFVV3,  FFHH3,  FFVV4, CFVV, CFHH
                , YY
        1000    FORMAT( 3F10.0,I5)
        1111    FORMAT(1H0, ' A = ',  F15.5 , 10X,  ' H = ', F15.5   //
                      1H0, ' THETA = ', F15.5, 15X, ' PRINT OPTION = ', I5    )
```

L.3 CONE, CYLINDER OR FRUSTRUM

A generalized expression of the far-field scattering from a cone, cylinder, or frustrum has been formulated using the Ruck-Ufimtsev technique (Ref. 3). The resulting expression of the scattered field is the following:

$$\left\{\sqrt{\sigma}\, e^{i\phi}\right\}_{\substack{V\\H}} = \mp\sqrt{\pi}\;\cdot$$

$$\Big\{-a_1 \Big[\big[J_2(\zeta_1)+iJ_1(\zeta_1)\big] C(n_-) \pm \big[J_o(\zeta_1)-iJ_1(\zeta_1)\big] B(n_-,\tfrac{\pi}{2}+\theta)\Big] Q(1)$$

$$-a_1 \Big[\big[J_2(\zeta_1)-iJ_1(\zeta_1)\big] C(n_-) \pm \big[J_o(\zeta_1)+iJ_1(\zeta_1)\big] B(n_-,\tfrac{\pi}{2}-\theta)\Big] Q(4)$$

$$-a_2 \Big[\big[J_2(\zeta_2)+iJ_1(\zeta_2)\big] C(n_+) \pm \big[J_o(\zeta_2)-iJ_1(\zeta_2)\big] B(n_+,\alpha+\theta)\Big] e^{i\psi}$$

$$-a_2 \Big[\big[J_2(\zeta_2)-iJ_1(\zeta_2)\big] C(n_+) \pm \big[J_o(\zeta_2)+iJ_1(\zeta_2)\big] B(n_+,\alpha-\theta)\Big] e^{i\psi} Q(3)$$

$$\pm a_1 \big[J_o(\zeta_1)-iJ_1(\zeta_1)\big]\tfrac{1}{2}\tan(\alpha+\theta)F(\tau_1)Q(1)$$

$$\pm a_1 \big[J_o(\zeta_1)+iJ_1(\zeta_1)\big]\tfrac{1}{2}\tan(\alpha-\theta)F(\tau_4)Q(34)$$

$$\mp a_2 \big[J_o(\zeta_2)-iJ_1(\zeta_2)\big]\tfrac{1}{2}\tan(\alpha+\theta)F(\tau_2)e^{i\psi}Q(1)$$

$$\mp a_2 \big[J_o(\zeta_2)+iJ_1(\zeta_2)\big]\tfrac{1}{2}\tan(\alpha-\theta)F(\tau_3)e^{i\psi}Q(34)\Big\}$$

and for $\pi/2 \leq \theta \leq \pi$,

$$\left\{\sqrt{\sigma}\, e^{i\phi}\right\}_{\substack{V\\H}} = \mp\sqrt{\pi} \quad .$$

$$\Big\{-a_1\Big[\big[J_2(\zeta_1)+iJ_1(\zeta_1)\big]C(n_-)\pm\big[J_o(\zeta_1)-iJ_1(\zeta_1)\big|\,B(n_-,\pi-\alpha-\theta)\Big]Q(1)$$

$$-a_2\Big[\big[J_2(\zeta_2)+iJ_1(\zeta_2)\big]C(n_+)\pm\big[J_o(\zeta_2)-iJ_1(\zeta_2)\big|\,B(n_+,\tfrac{3\pi}{2}-\theta)\Big]e^{i\psi}$$

$$-a_2\Big[\big[J_2(\zeta_2)-iJ_1(\zeta_2)\big]C(n_+)\pm\big[J_o(\zeta_2)+iJ_1(\zeta_2)\big|\,B(n_+,\theta-\tfrac{\pi}{2})\Big]e^{i\psi}Q(3)$$

$$\pm a_1\,\big|\,J_o(\zeta_1)-iJ_1(\zeta_1)\big|\tfrac{1}{2}\tan(\alpha+\theta)F(\tau_1)Q(1)$$

$$\mp a_2\,\big|\,J_o(\zeta_2)-iJ_1(\zeta_2)\big|\tfrac{1}{2}\tan(\alpha+\theta)F(\tau_2)e^{i\psi}Q(1)\ \Big\}$$

where the geometry of the problem is shown in Figure L.3-1 and

$$\begin{aligned}
C(n) &= \frac{1}{n}\sin\frac{\pi}{n}\left\{\cos\frac{\pi}{n}-1\right\}^{-1},\\
B(n,\phi) &= \frac{1}{n}\sin\frac{\pi}{n}\left\{\cos\frac{\pi}{n}-\cos\frac{2\phi}{n}\right\}^{-1},\\
\tau^2_{\binom{1}{2}} &= 2ka_{\binom{1}{2}}\csc\alpha\cos(\alpha+\theta),\\
\tau^2_{\binom{3}{4}} &= 2ka_{\binom{2}{1}}\csc\alpha\cos(\alpha-\theta),\\
\zeta_j &= 2ka_j\sin\theta,\\
\psi &= 2kh\cos\theta,\\
n_{\pm} &= \frac{3}{2}\pm\frac{\alpha}{\pi},\\
Q(1) &= Q(2ka_1(\pi-\alpha-\theta)),\\
k &= 2\pi/\lambda = \text{wave number},
\end{aligned}$$

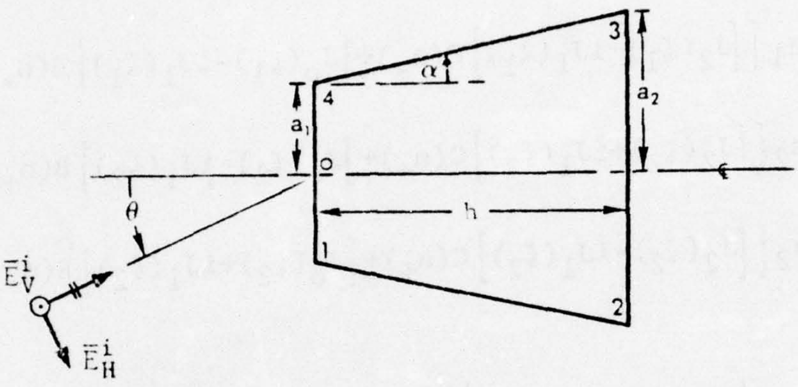


Fig. L.3-1 CONE, CYLINDER, OR FRUSTRUM SCATTERING GEOMETRY

$$Q(3) = Q(2ka_2(\alpha-\theta)(\theta-\pi/2)),$$

$$Q(34) = Q(2ka_2(\alpha-\theta)),$$

$$Q(4) = Q(2ka_1(\frac{\pi}{2}-\theta)).$$

The above solution to the scattering from a frustrum can simply be reduced to that of a right circular cylinder by setting F(τi) = 0 (i.e., $\alpha = 0$). For the case of a large right circular cone, it is important to note that numerical difficulty will be encountered if the above expressions were used to compute the scattered field near and at the conical specular aspect, $\theta = \pi/2 - \alpha$. An asymptotic solution to this problem can be expressed as

$$\lim_{\theta \to \pi/2-\alpha} \left\{\sqrt{\sigma}\, e^{i\phi}\right\}_{\substack{V\\H}} \simeq \mp \sqrt{\pi}\, a_2 e^{i\psi} \cdot$$

$$\left\{- \left[J_2(\zeta_2)+iJ_1(\zeta_2)\right] C(n_+) \pm \left[J_o(\zeta_2)-iJ_1(\zeta_2)\right] \times\right.$$

$$\left.\left[i\frac{2}{3}ka_2 \csc\alpha \sin(\alpha+\theta)\right]\right\} \cdot$$

The inputs, outputs, restrictions, and definition of key terms in the subroutine are presented in the following paragraphs.

L.3.1 Inputs

The subroutine inputs are read from cards or passed in a common block. The parameters passed in the common block include NMIN = minimum frequency number, NMAX = maximum frequency number, DF = frequency increment (in MHz) and FC = carrier frequency (in GHz). The inputs read from a card are the following:

	MATHEMATICAL			
	SYMBOL	VARIABLE	DEFINITION	COLUMNS
Card 1	a_1	A1	Smaller radius of frustrum (inches)	1-10
	a_2	A2	Larger radius of frustrum (inches)	11-20
	h_2	H2	Height of frustrum (inches)	21-30
		KONFIG	=1 Frustrum =2 Cone =3 Cylinder	31-35
	θ	THETAD	Azimuth angle (degrees)	36-45

L.3.2 Outputs

The data base output consists of four linear arrays, EVVR, EVVI, EHHR, EHHI, which contain the real and imaginary parts of the vertically and horizontally polarized back scattered fields (in meters) at frequency increments spaced DF MHz from NMIN*DF to NMAX*DF.

The frequency, vertically polarized cross section, and horizontally polarized cross section are printed out if the print option KP is positive.

L.3.3 Restrictions

L.3.3.1 <u>Physical Dimensions</u>

All target dimensions should be large with respect to the wavelength of the illuminating field.

L.3.3.2 Output

The output arrays are passed in the Argument List and a value is computed and stored only in locations NMIN to NMAX.

L.3.3.3 Azimuth

The azimuth angle is restricted to the region from 0 to 180 degrees. In addition, although the formulation is mathematically valid at specular points, the correct computational results are not provided at these angles. Thus a small angular offset should be used at angles of 0, 180, and ($\pi/2-\alpha$) degrees.

L.3.4 Definition of Sample Terms Used in Subroutine

$$\text{FSTCOX} = C(n) = \frac{1}{n} \sin \frac{\pi}{n}\left\{\cos \frac{\pi}{n} -1 \right\}^{-1}$$

Where $n_{\pm} = \frac{3}{2} \pm \frac{\alpha}{\pi}$

$$\text{FSTCON} = C(n_{-})$$

$$\text{FSTCOP} = C(n_{+})$$

$$\text{TAUSQ} \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} = 2ka_{\begin{pmatrix} 1 \\ 2 \end{pmatrix}} \text{CSC } \alpha$$

$$\text{XOPX11} = J_o(\xi_1)+iJ_1(\xi_1)$$

$$\text{XOPX12} = J_o(\xi_2)+iJ_1(\xi_2)$$

$$\text{Q1} = Q(1) = Q(2ka_1(\pi-\alpha-\theta)$$

$$\text{SECND 1} = B(n_{-}, \frac{\pi}{2} + \theta)$$

$$\text{WHERE} = B(n,\emptyset) = \frac{1}{n} \sin \frac{\pi}{n} \left\{\cos \frac{\pi}{\bar{n}} -\cos \frac{2\emptyset}{n}\right\} -1$$

$$\text{TERM 1} \begin{Bmatrix} V \\ H \end{Bmatrix} = -a_1 \left[\left[J_2(\zeta_1)+iJ_1(\zeta_1)\right] C(n_{-}) \pm \left[J_o(\zeta_1)-iJ_1(\zeta_1)\right] B(n_{-},\frac{\pi}{2}+\theta)\right] Q(1)$$

$$\text{TERM 4} \begin{Bmatrix} V \\ H \end{Bmatrix} = -a_1 \left[\left[J_2(\zeta_1)-iJ_1(\zeta_1)\right] C(n_{-}) \pm \left[J_o(\zeta_1)+iJ_1(\zeta_1)\right] B(n_{-},\frac{\pi}{2}-\theta)\right] Q(4)$$

$$\text{TERM 2} \begin{Bmatrix} V \\ H \end{Bmatrix} = -a_2 \left[\left[J_2(\zeta_2)+iJ_1(\zeta_2)\right] C(n_{+}) \pm \left[J_o(\zeta_2)-iJ_1(\zeta_2)\right] B(n_{+},\alpha+\theta)\right] e^{i\psi}$$

$$\text{TERM 3}\ \left\{\begin{matrix}V\\H\end{matrix}\right\} = -a_2\left[\left[J_2(\zeta_2)-iJ_1(\zeta_2)\right]C(n_+)\pm\left[J_o(\zeta_2)+iJ_1(\zeta_2)\right]B(n_+,\alpha-\theta)\right]e^{i\psi}Q(3)$$

$$\text{TERM 5}\ \left\{\begin{matrix}V\\H\end{matrix}\right\} = \pm a_1\left[J_o(\zeta_1)-iJ_1(\zeta_1)\right]\frac{1}{2}\tan(\alpha+\theta)F(\tau_1)Q(1)$$

WHERE = the upper sign is used for V polarization and the lower sign is used for H polarization

$$ZZ = \left[J_o(\zeta_2)-iJ_1(\zeta_2)\right]\cdot\left\{\frac{1}{2}\left[i\frac{2}{3}ka_2\csc\alpha\sin(\alpha+\theta)\right]\right\}$$

$$YY = -\left[J_2(\zeta_2)+iJ_1(\zeta_2)\right]C(n_+)$$

L.3.5 Subroutines Utilized

Subfunctions:

1. FIRS (XN) computes c(n)
2. SECO (PHI, XN) computes B (n, Ø)
3. Q (X) computes the Q function value with argument x
4. F (TAU) computes the F (τ) function value

Subroutines:

BESL (ARG1, XJO, XJ1, XJ2) returns

Jo (ARG1) in XJO
J1 (ARG1) in XJ1
J2 (ARG1) in XJ2

```
      SUBROUTINE TARGET ( EVVR, EVVI, EHHR, EHHI, THETAD)
C                                                                       *
C * * * * * GENERALIZED PROGRAM FOR A FRUSTRA, CONE, OR CYLINDER  * * *
C                   (UFIMTSEV SOLUTION FOR CW)                          *
C                                                                       *
C           A1      = FRONT-END RADIUS (INCHES)                         *
C           A2      = BACK-END RADIUS (SHOULD BE .GE. A1) - (INCHES)    *
C           H2      = TOTAL LENGTH (INCHES)                             *
C           ALPHA   = CONE OR FRUSTRA HALF ANGLE (DEGREES)              *
C           FREQ    = CARRIER FREQUENCY (GHZ)                           *
C           DELTHT  = ASPECT ANGLE INCREMENT (DEGREES) - (.GE. 0.1)     *
C           KONFIG  = TARGET CONFIGURATION                              *
C                        1 = FRUSTRA                                    *
C                        2 = CONE                                       *
C                        3 = CYLINDER                                   *
C                                                                       *
C * * * * ALL DIMENSIONS ARE IN INCHES AND ANGLES ARE IN DEGREES  * * *
C                                                                       *
      COMMON  MOVER, M, NMIN, NMAX, DF, FC, PW, T0
C     NMIN = MINIMUM FREQUENCY SAMPLE
C     NMAX = MAXIMUM FREQUENCY SAMPLE
C     DF   = FREQUENCY INCREMENT IN MHZ
C     FC   = CARRIER FREQUENCY IN GHZ
      DIMENSION  EVVR(512), EVVI(512), EHHR(512), EHHI(512),
     1           FREQ(512), SIGMAV(512), SIGMAH(512)
C
      COMPLEX PHASE, X0PX11, X0MX11, X0PX12, X0MX12, X2PX11, X2MX11,
     1        X2PX12, X2MX12, TERM1V, TERM1H, TERM2V, TERM2H, TERM3V,
     2      . TERM3H, TERM4V, TERM4H, TERM5V, TERM5H, TERM6V, TERM6H,
     3        TERM7V, TERM7H, TERM8V, TERM8H, FFVV, FFHH, F, FTAU1,
     4        FTAU2, FTAU3, FTAU4       ,XX,YY,ZZ
C                                                                       *
C                                                                       *
C                                                                       *
      READ(5,1000) A1,A2,H2, KONFIG, KP
 1000 FORMAT( 3F10.0,2I5 )
      ALP    = A2 - A1
      ALPHA  = ATAN2(ALP, H2)
      PI     = 3.14159265358979
      ANGLE  = (ALPHA * 180.0) / PI
C                                                                       *
      IF (KONFIG - 2) 30, 40, 50
C                                                                       *
   30 WRITE (6, 3030) H2, A1, A2, ANGLE
 3030 FORMAT (1H0///'          FRUSTRA (UFIMTSEV SOLUTION)'///'       FRUSTR
     1A LENGTH = ',F10.6,' INCHES'//'      FRONT-END RADIUS = ',F10.6,' INC
     2HES'//'       BACK-END RADIUS = ',F10.6,' INCHES'//' FRUSTRA HALF-ANG
     3LE = ',F10.6,' DEGREES')
      GO TO 60
C                                                                       *
   40 WRITE (6, 3040) H2, A2, ANGLE
 3040 FORMAT (1H0///'             CONE (UFIMTSEV SOLUTION)'///'         CONE
     1 LENGTH = ',F10.6,' INCHES'//'           BASE RADIUS = ',F10.6,' INCH
     2ES'//'      CONE HALF-ANGLE = ',F10.6,' DEGREES')
      GO TO 60
C                                                                       *
   50 WRITE (6, 3050) H2, A2
```

```
3050 FORMAT (1H0////          CYLINDER (UFIMTSEV SOLUTION)'///     CYLIND
    1ER LENGTH = ',F10.6,' INCHES'//     CYLINDER RADIUS = ',F10.6,' IN
    2CHES')
C                                                                        *
  60 WRITE (6, 79) THETAD
  79 FORMAT(1H0, ' THETA = ' , E15.5  /// )
C                                                                        *
     C      = 11.80285078
     DTR    = PI / 180.0
     PISQRT = SQRT(PI)
     PIOVR2 = PI / 2.0
     A12       = 2.0* A1
     A22       = 2.0* A2
     H22       = 2.0* H2
     XNPOS  = (3.0 / 2.0) + (ALPHA / PI)
     XNNEG  = (3.0 / 2.0) - (ALPHA / PI)
C    COMPUTE C(N)
     FSTCOP = FIRS(XNPOS)
     FSTCON = FIRS(XNNEG)
     THETA   = THETAD * DTR
     STHT    = SIN(THETA)
     CTHT    = COS(THETA)
     APT     = ALPHA + THETA
     AMT     = ALPHA - THETA
     TANAPT = TAN(APT)
     TANAMT = TAN(AMT)
     TWOPIC     = 2.0 * PI / C
C                                                                        *
     DO 200  I = NMIN, NMAX
     XI        = I - 1
     XK0       = TWOPIC * XI * DF / 1000.0
     X2KA1      = A12 * XK0
     X2KA2      = A22 * XK0
     X2KH2      = H22 * XK0
C                                                                        *
     IF (KONFIG .EQ. 3) GO TO 70
     TAUSQ1 = X2KA1 / SIN(ALPHA)
     TAUSQ2 = X2KA2 / SIN(ALPHA)
  70 CONTINUE
C                                                                        *
     WAPT    = PIOVR2 - ACOS(0.8 * COS(ALPHA) / X2KH2)
     ARG1    = X2KA1 * STHT
     ARG2    = X2KA2 * STHT
     FASE    = X2KH2 * CTHT
     PHASE   = CMPLX (COS(FASE), SIN(FASE))
C                                                                        *
     CALL BESL (ARG1, XJ01, XJ11, XJ21)
     CALL BESL (ARG2, XJ02, XJ12, XJ22)
C                                                                        *
     X0PX11 = CMPLX(XJ01, XJ11)
     X0MX11 = CMPLX(XJ01,-XJ11)
     X0PX12 = CMPLX(XJ02, XJ12)
     X0MX12 = CMPLX(XJ02,-XJ12)
     X2PX11 = CMPLX(XJ21, XJ11)
     X2MX11 = CMPLX(XJ21,-XJ11)
     X2PX12 = CMPLX(XJ22, XJ12)
     X2MX12 = CMPLX(XJ22,-XJ12)
                                                                         *
```

```
      Q1     = Q(X2KA1 * (PI - APT))
      Q4     = Q(X2KA1 * (PIOVR2 - THETA))
      Q3     = Q(X2KA2 * ((THETA+THETA-PI) * (-AMT)) / (PIOVR2-ALPHA))
      Q3A    = Q(X2KA2 * AMT)
C                                                                     *
      IF (THETA .GT. PIOVR2)  GO TO 90
C     DEFINE ANGLES FOR THET .LT. 90 DEGREES
C                                                                     *
      PHI1   = PIOVR2 + THETA
      PHI4   = PIOVR2 - THETA
      PHI2   = APT
      GO TO 98
C     DEFINE ANGLES FOR THET .GT. 90 DEGREES
   90 PHI1   = PI-APT
      PHI4   = 0.0
      PHI2   = (3.0 * PIOVR2) - THETA
C                                                                     *
   98 IF ( Q3A .EQ. 0.0 ) GO TO 99
      PHI3 = AMT
      GO TO 100
   99 PHI3 = PIOVR2 - THETA
C                                                                     *
C     COMPUTE B(N,PHI)
  100 SECND1 = SECO(PHI1, XNNEG)
      SECND4 = SECO(PHI4, XNNEG)
      SECND2 = SECO(PHI2, XNPOS)
      SECND3 = SECO(PHI3, XNPOS)
C                                                                     *
      TERM1V = 0.0
      TERM1H = 0.0
      TERM4V = 0.0
      TERM4H = 0.0
      TERM5V = 0.0
      TERM5H = 0.0
      TERM6V = 0.0
      TERM6H = 0.0
      TERM7V = 0.0
      TERM7H = 0.0
      TERM8V = 0.0
      TERM8H = 0.0
C                                                                     *
      IF (KONFIG .EQ. 2) GO TO 120
C                                                                     *
      TERM1V = -A1 * ((X2PX11 * FSTCON) + (X0MX11 * SECND1)) * Q1
      TERM1H = -A1 * ((X2PX11 * FSTCON) - (X0MX11 * SECND1)) * Q1
      TERM4V = -A1 * ((X2MX11 * FSTCON) + (X0PX11 * SECND4)) * Q4
      TERM4H = -A1 * ((X2MX11 * FSTCON) - (X0PX11 * SECND4)) * Q4
C                                                                     *
  120 TERM2V = -A2 * ((X2PX12 * FSTCOP) + (X0MX12 * SECND2)) * PHASE
      TERM2H = -A2 * ((X2PX12 * FSTCOP) - (X0MX12 * SECND2)) * PHASE
      TERM3V = -A2 * ((X2MX12 * FSTCOP) + (X0PX12 * SECND3)) * Q3*PHASE
      TERM3H = -A2 * ((X2MX12 * FSTCOP) - (X0PX12 * SECND3)) * Q3*PHASE
C                                                                     *
      IF (KONFIG .EQ. 3) GO TO 130
      IF (KONFIG .EQ. 2) GO TO 140
C                                                                     *
C     THIS SECTION NOT DONE FOR CYLINDER OR CONE
      TAU1   = TAUS01 * COS(APT)
      TAU4   = TAUSQ1 * COS(AMT)
      FTAU1  = F(TAU1)
      FTAU4  = F(TAU4)
      TERM5V = A1 * TANAPT * X0MX11 * FTAU1 * 0.5 * Q1
      TERM5H = -TERM5V
      TERM6V = A1 * TANAMT * X0PX11 * FTAU4 * 0.5 * Q3A
      TERM6H = -TERM6V
C     THIS SECTION NOT DONE FOR CYLINDER
C                    DONE FOR CONE
  140 TAU2   = TAUSQ2 * COS(APT)
      TAU3   = TAUSQ2 * COS(AMT)
      FTAU2  = F(TAU2)
      FTAU3  = F(TAU3)
      TERM7V =-A2 * TANAPT * X0MX12 * FTAU2 * 0.5 * Q1 * PHASE
      TERM7H = -TERM7V
      TERM8V =-A2 * TANAMT * X0PX12 * FTAU3 * 0.5 * Q3A* PHASE
      TERM8H = -TERM8V
C                                                                      *
      IF (KONFIG - 2) 130, 56, 130
   56 IF (ABS(APT-PIOVR2) .GT. WAPT ) GO TO 130
C     THIS SECTION USED TO COMPUTE CONE RETURN NEAR SPECULAR TO
C                    CONIC SURFACE  (THET NEAR (PI/2)-ALPHA)
      XX        = CMPLX(0.0,1.0)
      ZZ        = X0MX12 * XX * TAUSQ2 * SIN(APT) /3.0
      YY        =-X2PX12 * FSTCOP
      FFVV      =-PISQRT * A2 * PHASE * ( YY + ZZ)
      FFHH      = PISQRT * A2 * PHASE * ( YY - ZZ)
      GO TO 55
C
  130 FFVV     = -        PISQRT * (TERM1V + TERM2V + TERM3V + TERM4V +
     1                              TERM5V + TERM6V + TERM7V + TERM8V )
      FFHH     =          PISQRT * (TERM1H + TERM2H + TERM3H + TERM4H +
     1                              TERM5H + TERM6H + TERM7H + TERM8H )
C                                                                      *
   55 CONTINUE
      FFVV       = FFVV * 0.02540005
      FFHH       = FFHH * 0.02540005
C
      EVVR(I) = REAL (FFVV)
      EVVI(I) =-AIMAG(FFVV)
      EHHR(I) = REAL (FFHH)
      EHHI(I) =-AIMAG(FFHH)
C                                                                      *
      IF (KP)  77, 77, 78
   78 FREQ(I)   = XI * DF / 1000.0
      SIGMAV(I)= 10.0*ALOG10(EVVR(I)*EVVR(I) + EVVI(I)*EVVI(I))
      SIGMAH(I)= 10.0*ALOG10(EHHR(I)*EHHR(I) + EHHI(I)*EHHI(I))
   77 CONTINUE
C                                                                      *
  200 CONTINUE
C                                                                      *
C                                                                      *
C     IF (KP)  75, 75, 76
C  76 WRITE (6, 74) ( FREQ(I), SIGMAV(I), SIGMAH(I), I= NMIN, NMAX )
C  74 FORMAT (1H0, 3E15.5 )
   75 CONTINUE
C                                                                      *
C
```

```
      RETURN
      END
      FUNCTION FIRS(XN)
      PI      = 3.14159265358979
      A       = (SIN(PI / XN)) / XN
      B       = 1.0 / (COS(PI / XN) - 1.0)
      FIRS    = A * B
      RETURN
      END
      FUNCTION SECO(PHI, XN)
      PI      = 3.14159265358979
      A       = (SIN(PI / XN)) / XN
      B       = COS(PI / XN) - COS((2.0 * PHI)/ XN)
      C       = 1.0 / B
      SECO    = A * C
      RETURN
      END
      FUNCTION Q(Z)
C     Q(Z) = 0.5*(1 + ERF(Z))
C   * ERF(Z) IS EVALUATED USING A RATIONAL POLYNOMIAL APPROXIMATION
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,
C   *             SECTION 7.1.26)
C
      IF ( Z.GT. 2.) GO TO 10
      IF ( Z.LT.-2.) GO TO 20
      AZ = ABS(Z)
      P = 1.0/(1.0 + .47047*AZ)
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)
      IF (Z) 2,4,6
    2 Q = (1.0 - Y)/2.
      RETURN
    4 Q = .5
      RETURN
    6 Q = (1.0 + Y)/2.
      RETURN
   10 Q = 1.
      RETURN
   20 Q = 0.
      RETURN
      END
      SUBROUTINE BESL ( X, B0, B1, B2 )
C
C   * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS
C   * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS
C   * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )
C
      S = 1.0
      IF (X .LT. 0.0) S = -1.0
      X = ABS (X)
C
      IF ( X .GT. 1.E-6 ) GO TO 5
      B0 = 1.0
     B1 = 0.0
      B2 = 0.0
      X = X * S
      RETURN
C
    5 CONTINUE
```

```
C
    1 IF ( X .GE. 3.) GO TO 9
      X1 = X/3.
      X1 = X1*X1
      B = 1. + X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479
     1 +       X1*(-.0039444+ X1*2.1E-4 )))) )
      GO TO 10
C
    9 X2 = 3./X
      F0 =  .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*
     1      (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )
      T0 = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573
     1     +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))
      B = F0*COS(T0)/SQRT(X)
C
   10 B0 = B
C
    2 IF ( X .GE. 3. ) GO TO 19
      X1 = X/3.
      X1 = X1*X1
      B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*
     1                (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )
      GO TO 20
C
   19 X2 = 3./X
      F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*
     1      (-.00249511 +X2*(.00113653 -.00020033*X2 )))))
      T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879
     1     +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))
      B = F1*COS(T1)/SQRT(X)
C
   20 B1 = B * S
      X = X * S
      B2= (2./X)*B1 - B0
   50 RETURN
      END
      COMPLEX FUNCTION F(TAU)
C
C     COMPUTES FTAU WHERE FTAU =(EXP(-J*TAU**2)/2*TAU)*SQRT(PI/2.)*
C                                    (C2(TAU**2) + J*S2(TAU**2))
C
      COMPLEX B, FP
      PI = 3.14159265358979
      PIO2 = PI/2.
      C1 = SQRT(PI/2.)
      C2 = 1./C1
      ATAUS = ABS(TAU)
      IF (ATAUS .LE. 0.5  )GO TO 20
C
C     FOR TAUS .GT. 0.5, FUNCTION COMPUTED USING POLYNOMIAL
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,
C   *            SECTIONS  7.3.9,7.3.10,7.3.32,7.3.33)
      TAUS = SQRT(ATAUS)
      X = C2*TAUS
      XS = X*X
C
      FX = (1.0+0.926*X)/(2.0+1.792*X+3.104*XS)
      GX = 1.0/(2.0+4.142*X+3.492*XS+6.67*X*XS)
```

```
C
      CC1XS = COS(ATAUS)
      SC1XS = SIN(ATAUS)
C
      CX = 0.5 + FX*SC1XS - GX*CC1XS
      SX = 0.5 - FX*CC1XS - GX*SC1XS
C
      IF (TAU .LT. 0.0) GO TO 10
      B = CMPLX(CX,SX)
      FP = CMPLX(  COS(ATAUS), -SIN(ATAUS) )
      F = (C1*B*FP)/TAUS
      RETURN
C
   10 CONTINUE
      B = CMPLX(SX,CX)
      A = ATAUS-PIO2
      FP = CMPLX( COS(A),SIN(A) )
      F = (B*FP*C1)/TAUS
      RETURN
C
   20 CONTINUE
C     FOR TAUS .LE. 0.5, FUNCTION IS EXPANDED IN SERIES AND FIRST FEW
C                        TERMS INTEGRATED TERM BY TERM TO OBTAIN RESULT
      FP = CMPLX(COS(TAU),-SIN(TAU))
      TS = TAU*TAU
      FR = 1 - TS*( .1 - .0046296296*TS)
      FI = TAU *( .333333333 - TS*(.0238095238 - 7.57575757E-4*TS))
      B = CMPLX(FR,FI)
      F = FP*B
      RETURN
      END
```

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI,THETAD)

L-38

```
      -----------
     / TARGET  /
    -----------

          *

  *
  * * * * *
 GENERALIZED PROGRAM                                            *
 FOR A FRUSTRA, CONE,
 OR CYLINDER  * * *                                            09.06--->*
 (UFIMTSEV SOLUTION                                              40     |       09
 FOR CW)                                                          ---------------
  *                                                              / WRITE TO DEV  /
  *                                                             /       6       /
 A1      = FRONT-END                                           /  VIA FORMAT   /
 RADIUS (INCHES)                                              /     3040      /
  *                                                          / FROM THE LIST /
 A2      = BACK-END                                          ---------------
 RADIUS (SHOULD BE                                                  |
 .GE. A1) - (INCHES)                                                |
  *                                                                 | NOTE  10
 H2      = TOTAL LENGTH                                      * * * * * * * * * * * *
 (INCHES)                                                    *  LIST = H2, A2,     *
  *                                                          *       ANGLE         *
 ALPHA  = CONE OR                                            * * * * * * * * * * * *
 FRUSTRA HALF ANGLE                                                 |
 (DEGREES)                                                          |
  *                                                                 |
 FREC    = CARRIER                                                 ...
 FREQUENCY (GHZ)                                                  .   .
  *                                                              .10.03.
 DELTHT = ASPECT ANGLE                                            .   .
 INCREMENT (DEGREES) -                                             ... 60
 (.GE. 0.1)          *
 KONFIG = TARGET
 CONFIGURATION
  *
 1 = FRUSTRA
  *
 2 = CONE
  *
 3 = CYLINDER
  *
  *
  * * * * ALL
 DIMENSIONS ARE IN
 INCHES AND ANGLES ARE
 IN DEGREES  * * *


 NMIN = MINIMUM
 FREQUENCY SAMPLE
 NMAX = MAXIMUM
```

2

NMIN = MINIMUM
FREQUENCY SAMPLE
NMAX = MAXIMUM
FREQUENCY SAMPLE
DF = FREQUENCY
INCREMENT IN MHZ
FC = CARRIER
FREQUENCY IN GHZ

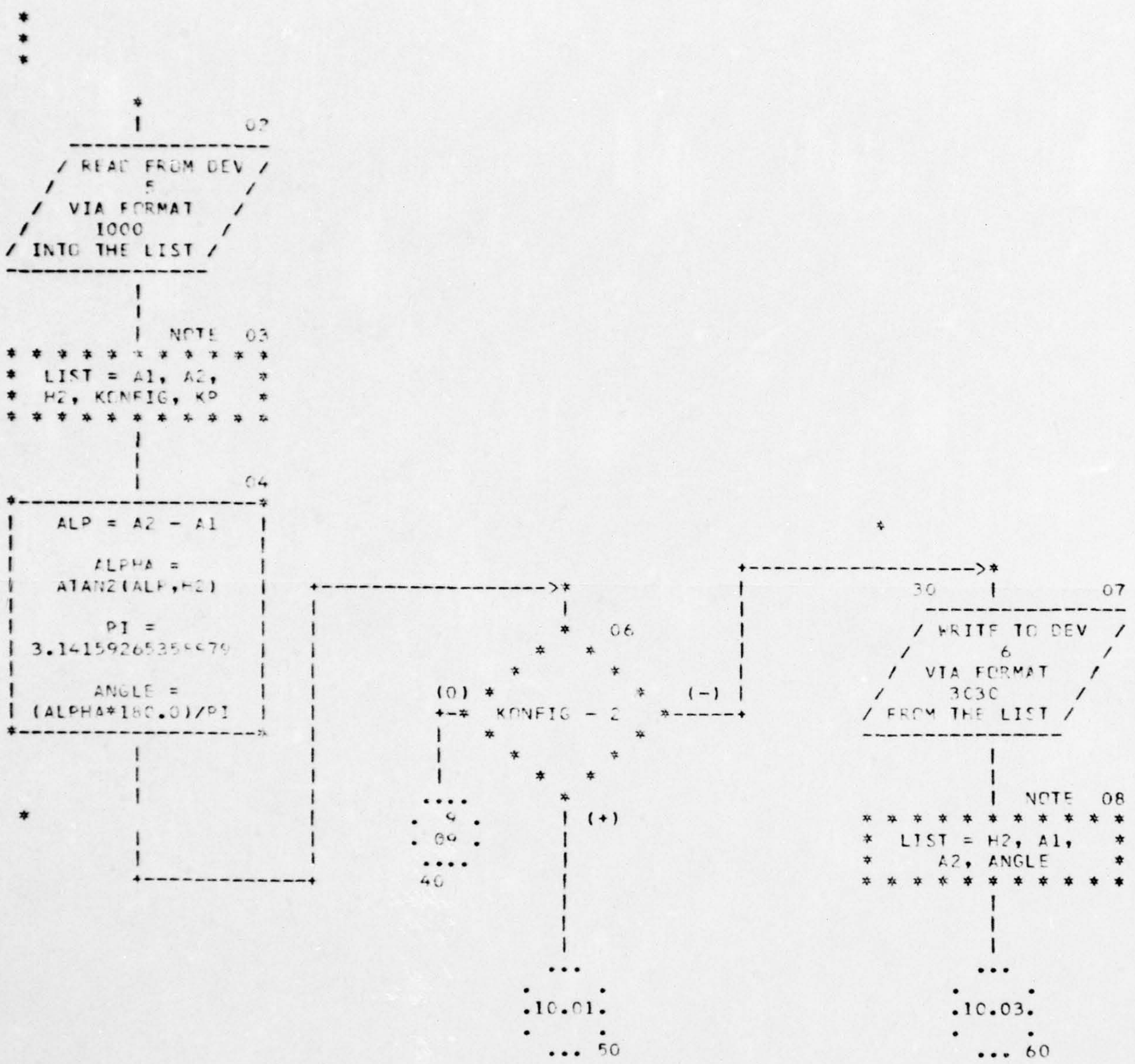


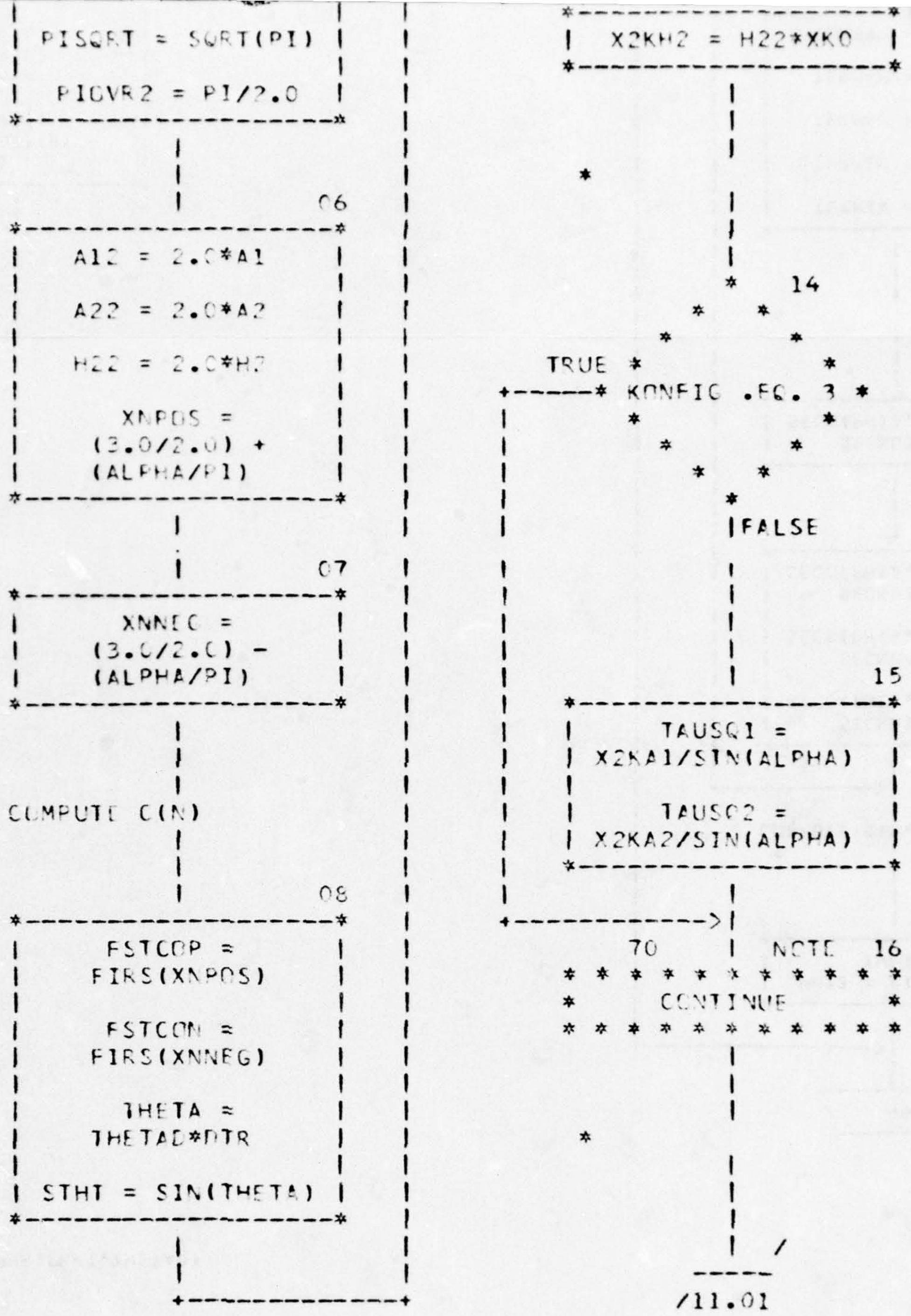
CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI,THETAD)

```
                        +----------------->*
                        |                  |       09
 *                      |        *-------------------*
                        |        | CTHT = COS(THETA) |
 09.06--->*             |        |                   |
    50    |        01   |        |   APT = ALPHA +   |
     ----------------   |        |       THETA       |
    / WRITE TO DEV  /   |        |                   |
   /       6       /    |        |   AMT = ALPHA -   |
  /  VIA FORMAT   /     |        |       THETA       |
 /     3050      /      |        |                   |
/ FROM THE LIST /       |        | TANAPT = TAN(APT) |
----------------        |        *-------------------*
          |             |                  |
          |             |                  |
          | NOTE  02    |                  |       10
* * * * * * * * * * *   |        *-------------------*
*   LIST = H2, A2   *   |        | TANAMT = TAN(AMT) |
* * * * * * * * * * *   |        |                   |
          |             |        | TWOPIC = 2.0*PI/C |
          |             |        *-------------------*
          |             |                  |
 *                      |                  |
          |             |                  |
 09.08*-->|             |         *        |
    60    |        03   |                  |
     ----------------   |                  |
    / WRITE TO DEV  /   |                  |  NOTE  11
   /       6       /    |        * * * * * * * * * * *
  /  VIA FORMAT   /     |        *   BEGIN DO LOOP   *
 /       79      /      |        *   200 I = NMIN,   *
/ FROM THE LIST /       |        *       NMAX        *
----------------        |        * * * * * * * * * * *
          |             |                  |
          |             |         13.10--->|
```

2

```
  ---------------
        |
        |  NOTE   04
 * * * * * * * * * * *
 *   LIST = THETAD   *
 * * * * * * * * * * *
        |
        |
        |
 *
        |
        |             05
 *-------------------*
 | C = 11.80285078   |
 |                   |
 | DTR = PI/180.0    |
 |                   |
 | PISQRT = SQRT(PI) |
 |                   |
 | PIOVR2 = PI/2.0   |
 *-------------------*
        |
        |
        |             06
 *-------------------*
 |  A12 = 2.0*A1     |
 |                   |
 |  A22 = 2.0*A2     |
 |                   |
 |  H22 = 2.0*H2     |
 |                   |
 |    XNPOS =        |
 |  (3.0/2.0) +      |
 |  (ALPHA/PI)       |
 *-------------------*
        |
        |             07
 *-------------------*
 |    XNNEG =        |
 |  (3.0/2.0) -      |
 |  (ALPHA/PI)       |
 *-------------------*
        |
        |
 COMPUTE C(N)
        |
        |             08
 *-------------------*
 |    FSTCOP =       |
 |   FIRS(XNPOS)     |
```

```
 * * * * * * * * * * *
            |
   13.10--->|
            |        12
 *-------------------*
 |   XI = I - 1      |
 |                   |
 |     XKO =         |
 |  TWOPIC*XI*DF/    |
 |     1000.0        |
 |                   |
 | X2KA1 = A12*XKO   |
 |                   |
 | X2KA2 = A22*XKO   |
 *-------------------*
            |
            |
            |        13
 *-------------------*
 | X2KH2 = H22*XKO   |
 *-------------------*
            |
            |
            |
 *
            |
            |
            |
            *  14
          *   *
        *       *
 TRUE  *          *
 +-----* KONFIG .EQ. 3 *
 |     *          *
 |       *      *
 |         *  *
 |          *
 |          |FALSE
 |          |
 |          |
 |          |
 |          |        15
 |  *-------------------*
 |  |    TAUSQ1 =       |
 |  | X2KA1/SIN(ALPHA)  |
 |  |                   |
 |  |    TAUSQ2 =       |
 |  | X2KA2/SIN(ALPHA)  |
 |  *-------------------*
 |          |
 +--------->|
      70    |  NOTE  16
 * * * * * * * * * * *
 *     CONTINUE      *
```


PISQRT = SQRT(PI)
PIOVR2 = PI/2.0
06
A12 = 2.0*A1
A22 = 2.0*A2
H22 = 2.0*H2
XNPOS =
(3.0/2.0) +
(ALPHA/PI)
07
XNNEG =
(3.0/2.0) -
(ALPHA/PI)
COMPUTE C(N)
08
FSTCOP =
FIRS(XNPOS)
FSTCON =
FIRS(XNNEG)
THETA =
THETAD*DTR
STHT = SIN(THETA)
X2KH2 = H22*XKO
14
TRUE
KONFIG .EQ. 3
FALSE
15
TAUSQ1 =
X2KA1/SIN(ALPHA)
TAUSQ2 =
X2KA2/SIN(ALPHA)
70
NOTE 16
CONTINUE
/11.01

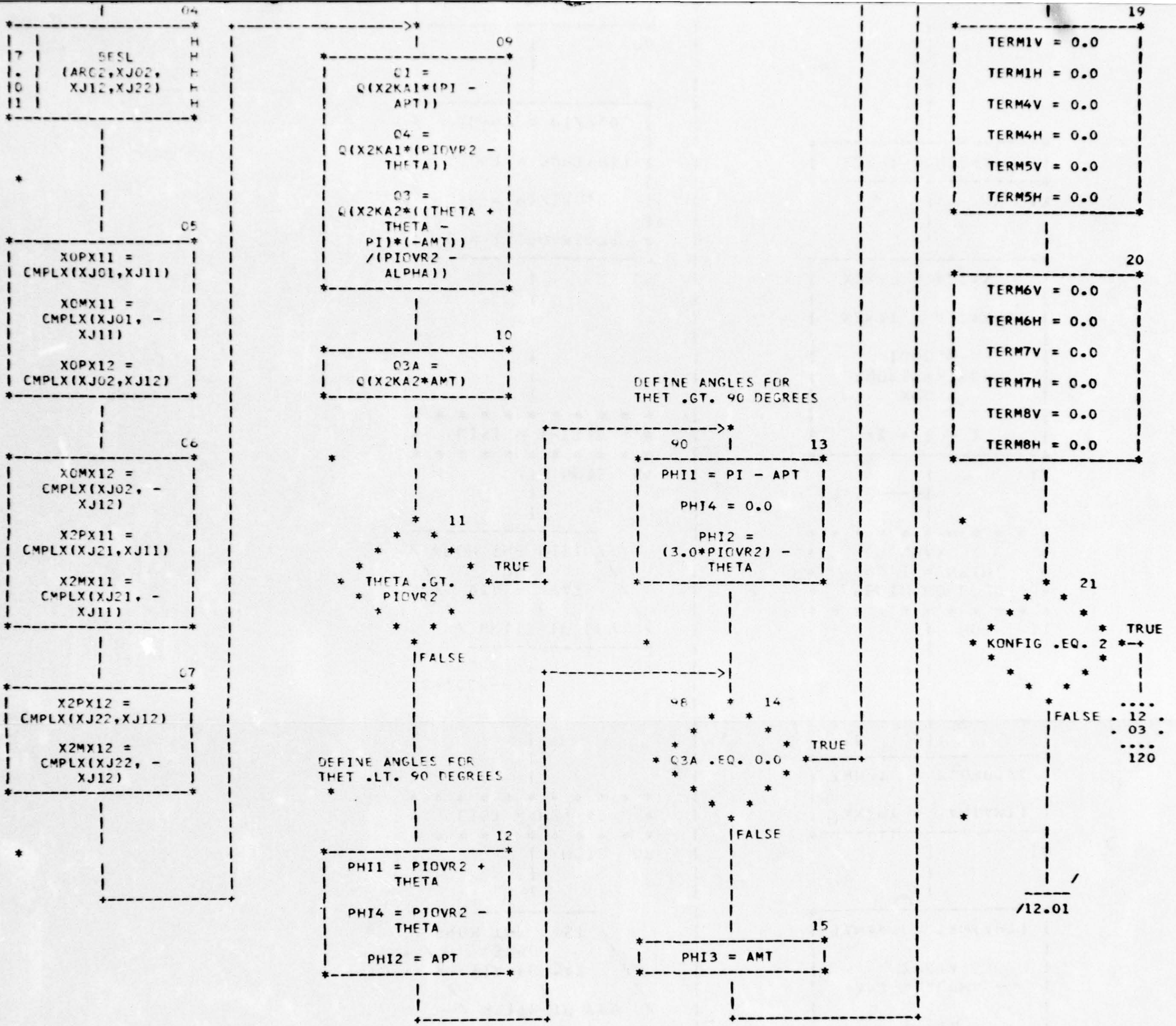
L-39

3

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI,THETAD)

L-40

```
                                                                                                            ------------
                                                                                                            /    99    /
                                                                                                            ------------
                                                                                                                 |
   10.17--->*                                                                                *------------------>|
            |        01                                                                      |                   |        16
  *-------------------*                                                                      |         *-------------------*
  |  WAPT = PIOVR2 -  |                                                                      |         |  PHI3 = PIOVR2 -  |
  |   ACOS(0.8*COS    |                                                                      |         |       THETA       |
  |  (ALPHA)/X2KH2)   |                                                                      |         *-------------------*
  |                   |                                                                      |                   |
  | ARC1 = X2KA1*STHT |                                                                      |                   |
  |                   |                                                                      |                   |
  | ARC2 = X2KA2*STHT |                                                                      |          *
  *-------------------*                                                                      |         COMPUTE B(N,PHI)
            |                                                                                |                   |
            |                                                                                |     *------------>|
            |        02                                                                      |     |      100    |        17
  *-------------------*                                                                      |     |   *-------------------*
  | FASE = X2KH2*CTHT |                                                                      |     |   |     SECND1 =      |
  |                   |                                                                      |     |   | SECO(PHI1,XNNEG)  |
  |      PHASE =      |                                                                      |     |   |                   |
  | CMPLX(COS(FASE),  |                                                                      |     |   |     SECND4 =      |
  |    SIN(FASE))     |                                                                      |     |   | SECO(PHI4,XNNEG)  |
  *-------------------*                                                                      |     |   |                   |
            |                                                                                |     |   |     SECND2 =      |
            |                                                                                |     |   | SECO(PHI2,XNPOS)  |
            |                                                                                |     |   *-------------------*
   *                                                                                         |     |             |
            |                                                                                |     |             |
            |                                                                                |     |             |        18
            |        03                                                                      |     |   *-------------------*
  *-------------------*                                                                      |     |   |     SECND3 =      |
  |  |                H                                                                      |     |   | SECO(PHI3,XNPOS)  |
  |7 |     BESL       H                                                                      |     |   *-------------------*
  |. |  (ARG1,XJ01,   H                                                                      |     |             |
  |0 |   XJ11,XJ21)   H                                                                      |     |             |
  |1 |                H                                                                      |     |             |
  *-------------------*                                                                      |     |    *
            |                                                                                |     |             |
            |                                                                                |     |             |
            |        04                                                                      |     |             |        19
  *-------------------*   *------------------>*                                              |     |   *-------------------*
  |  |                H   |                   |        09                                    |     |   |   TERM1V = 0.0    |
  |7 |     BESL       H   |         *-------------------*                                    |     |   |                   |
  |. |  (ARG2,XJ02,   H   |         |       Q1 =        |                                    |     |   |   TERM1H = 0.0    |
  |0 |   XJ12,XJ22)   H   |         |   Q(X2KA1*(PI -   |                                    |     |   |                   |
  |1 |                H   |         |       APT))       |                                    |     |   |   TERM4V = 0.0    |
  *-------------------*   |         |                   |                                    |     |   |                   |
            |             |         |       Q4 =        |                                    |     |   |   TERM4H = 0.0    |
            |             |         | Q(X2KA1*(PIOVR2 - |                                    |     |   |                   |
            |             |         |      THETA))      |                                    |     |   |   TERM5V = 0.0    |
   *                      |         |                   |                                    |     |   |                   |
            |             |         |       Q3 =        |                                    |     |   |   TERM5H = 0.0    |
            |             |         | Q(X2KA2*((THETA + |                                    |     |   *-------------------*
            |        05   |         |      THETA -      |                                    |     |             |
```

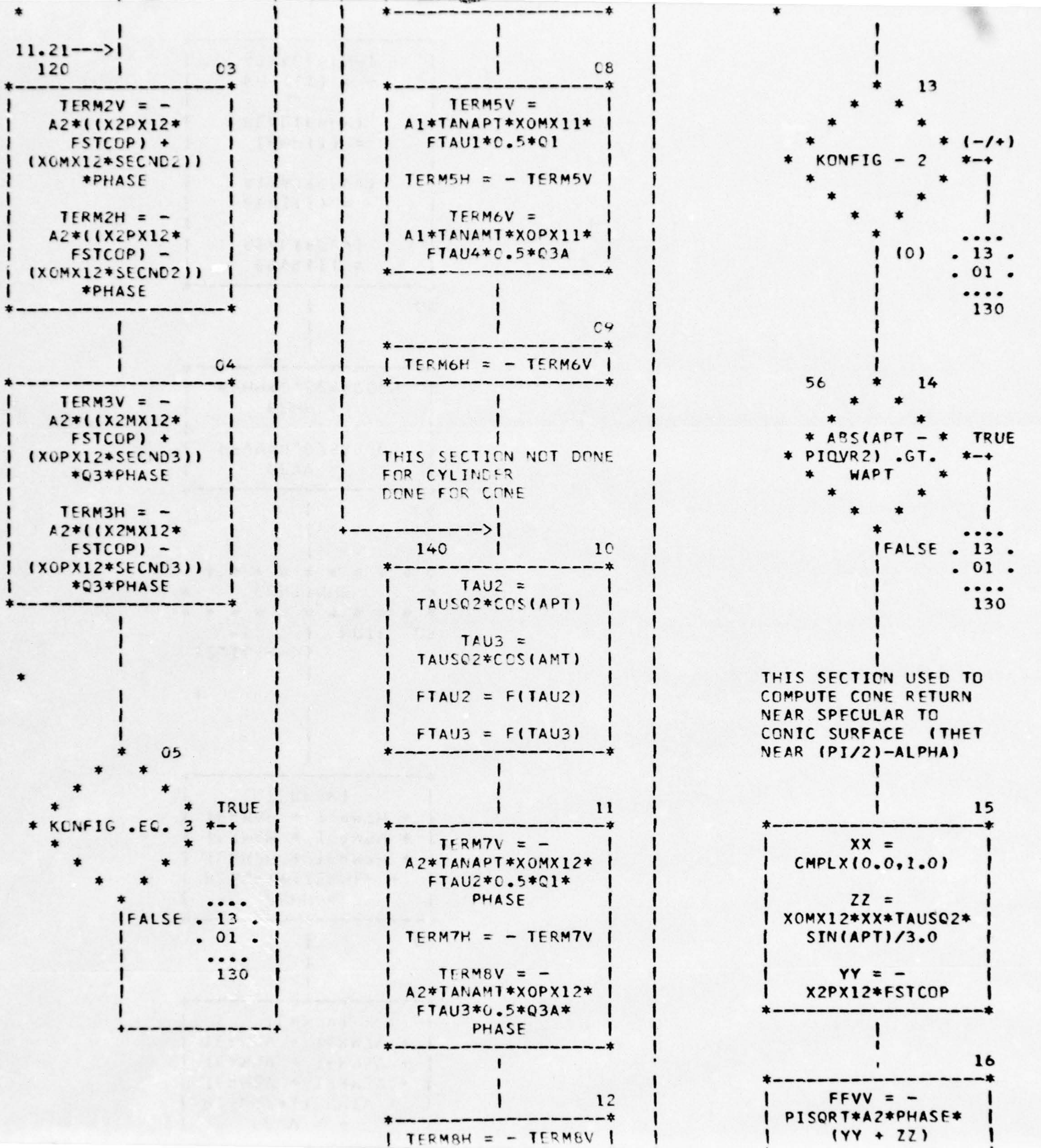

04
7.01
SESL (ARG2,XJ02, XJ12,XJ22)
05
X0PX11 = CMPLX(XJ01,XJ11)
X0MX11 = CMPLX(XJ01, - XJ11)
X0PX12 = CMPLX(XJ02,XJ12)
06
X0MX12 = CMPLX(XJ02, - XJ12)
X2PX11 = CMPLX(XJ21,XJ11)
X2MX11 = CMPLX(XJ21, - XJ11)
07
X2PX12 = CMPLX(XJ22,XJ12)
X2MX12 = CMPLX(XJ22, - XJ12)
09
Q1 = Q(X2KA1*(PI - APT))
Q4 = Q(X2KA1*(PIOVR2 - THETA))
Q3 = Q(X2KA2*((THETA + THETA - PI)*(-AMT)) /(PIOVR2 - ALPHA))
10
Q3A = Q(X2KA2*AMT)
11
THETA .GT. PIOVR2
TRUE
FALSE
DEFINE ANGLES FOR THET .LT. 90 DEGREES
12
PHI1 = PIOVR2 + THETA
PHI4 = PIOVR2 - THETA
PHI2 = APT
DEFINE ANGLES FOR THET .GT. 90 DEGREES
90
13
PHI1 = PI - APT
PHI4 = 0.0
PHI2 = (3.0*PIOVR2) - THETA
98
14
Q3A .EQ. 0.0
TRUE
FALSE
15
PHI3 = AMT
19
TERM1V = 0.0
TERM1H = 0.0
TERM4V = 0.0
TERM4H = 0.0
TERM5V = 0.0
TERM5H = 0.0
20
TERM6V = 0.0
TERM6H = 0.0
TERM7V = 0.0
TERM7H = 0.0
TERM8V = 0.0
TERM8H = 0.0
21
KONFIG .EQ. 2
TRUE
FALSE
12
03
120
/12.01

2

CHART TITLF - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI,THETAD)

```
 11.22--->*             +------------------->*
          |        01   |                    |
*-------------------*   |                    *  06
|    TERM1V = -     |   |                 *     *
|   A1*((X2PX11*    |   |              *           *
|     FSTCON) +     |   |        TRUE *               *
| (XOMX11*SECND1))  |   |    +-----* KONFIG .EQ. 2 *
|        *Q1        |   |    |        *               *
|                   |   |    |           *         *
|    TERM1H = -     |   |    |              *   *
|   A1*((X2PX11*    |   |    |                 *
|     FSTCON) -     |   |    |                 |FALSE
| (XOMX11*SECND1))  |   |    |                 |
|        *Q1        |   |    |                 |
*-------------------*   |    |                 |
          |             |    |                 |
          |        02   |    |                 |
*-------------------*   |    |   *
|    TERM4V = -     |   |    |   THIS SECTION NOT DONE
|   A1*((X2MX11*    |   |    |   FOR CYLINDER OR CONE
|     FSTCON) +     |   |    |                 |
| (XOPX11*SECND4))  |   |    |                 |        07
|        *Q4        |   |    |   *-------------------------*
|                   |   |    |   |        TAU1 =           |
|    TERM4H = -     |   |    |   |  TAUSQ1*COS(APT)        |
|   A1*((X2MX11*    |   |    |   |                         |
|     FSTCON) -     |   |    |   |        TAU4 =           |
| (XOPX11*SECND4))  |   |    |   |  TAUSQ1*COS(AMT)        |
|        *Q4        |   |    |   |                         |
*-------------------*   |    |   |  FTAU1 = F(TAU1)        |
          |             |    |   |                         |   +-------------------->*
          |             |    |   |  FTAU4 = F(TAU4)        |   |                     |
 *        |             |    |   *-------------------------*   |         *           |
          |             |    |                 |               |                     |
 11.21--->|             |    |                 |               |                     |
  120     |        03   |    |                 |        08     |                     *  13
*-------------------*   |    |   *-------------------------*   |                  *     *
|    TERM2V = -     |   |    |   |       TERM5V =          |   |               *           *
|   A2*((X2PX12*    |   |    |   |  A1*TANAPT*XOMX11*      |   |
```

2



11.21--->
120
03
TERM2V = -
A2*((X2PX12*
FSTCOP) +
(XOMX12*SECND2))
*PHASE
TERM2H = -
A2*((X2PX12*
FSTCOP) -
(XOMX12*SECND2))
*PHASE
04
TERM3V = -
A2*((X2MX12*
FSTCOP) +
(XOPX12*SECND3))
*Q3*PHASE
TERM3H = -
A2*((X2MX12*
FSTCOP) -
(XOPX12*SECND3))
*Q3*PHASE
05
KONFIG .EQ. 3
TRUE
FALSE
13
01
130
08
TERM5V =
A1*TANAPT*XOMX11*
FTAU1*0.5*Q1
TERM5H = - TERM5V
TERM6V =
A1*TANAMT*XOPX11*
FTAU4*0.5*Q3A
09
TERM6H = - TERM6V
THIS SECTION NOT DONE
FOR CYLINDER
DONE FOR CONE
140
10
TAU2 =
TAUSQ2*COS(APT)
TAU3 =
TAUSQ2*COS(AMT)
FTAU2 = F(TAU2)
FTAU3 = F(TAU3)
11
TERM7V = -
A2*TANAPT*XOMX12*
FTAU2*0.5*Q1*
PHASE
TERM7H = - TERM7V
TERM8V = -
A2*TANAMT*XOPX12*
FTAU3*0.5*Q3A*
PHASE
12
TERM8H = - TERM8V
13
KONFIG - 2
(-/+)
(0)
13
01
130
56
14
ABS(APT -
PIQVR2) .GT.
WAPT
TRUE
FALSE
13
01
130
THIS SECTION USED TO
COMPUTE CONE RETURN
NEAR SPECULAR TO
CONIC SURFACE (THET
NEAR (PI/2)-ALPHA)
15
XX =
CMPLX(0.0,1.0)
ZZ =
XOMX12*XX*TAUSQ2*
SIN(APT)/3.0
YY = -
X2PX12*FSTCOP
16
FFVV = -
PISQRT*A2*PHASE*
(YY + ZZ)

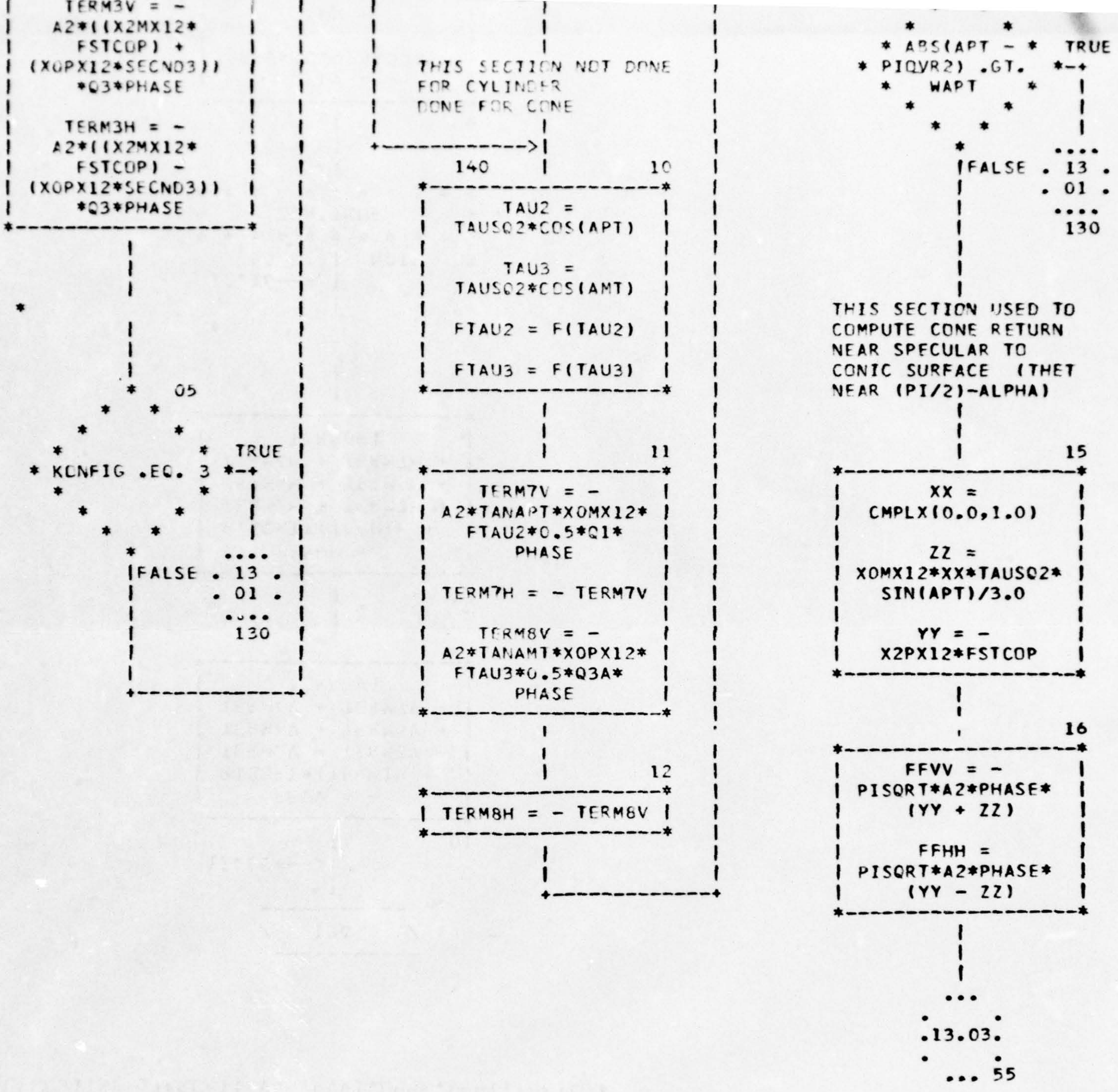
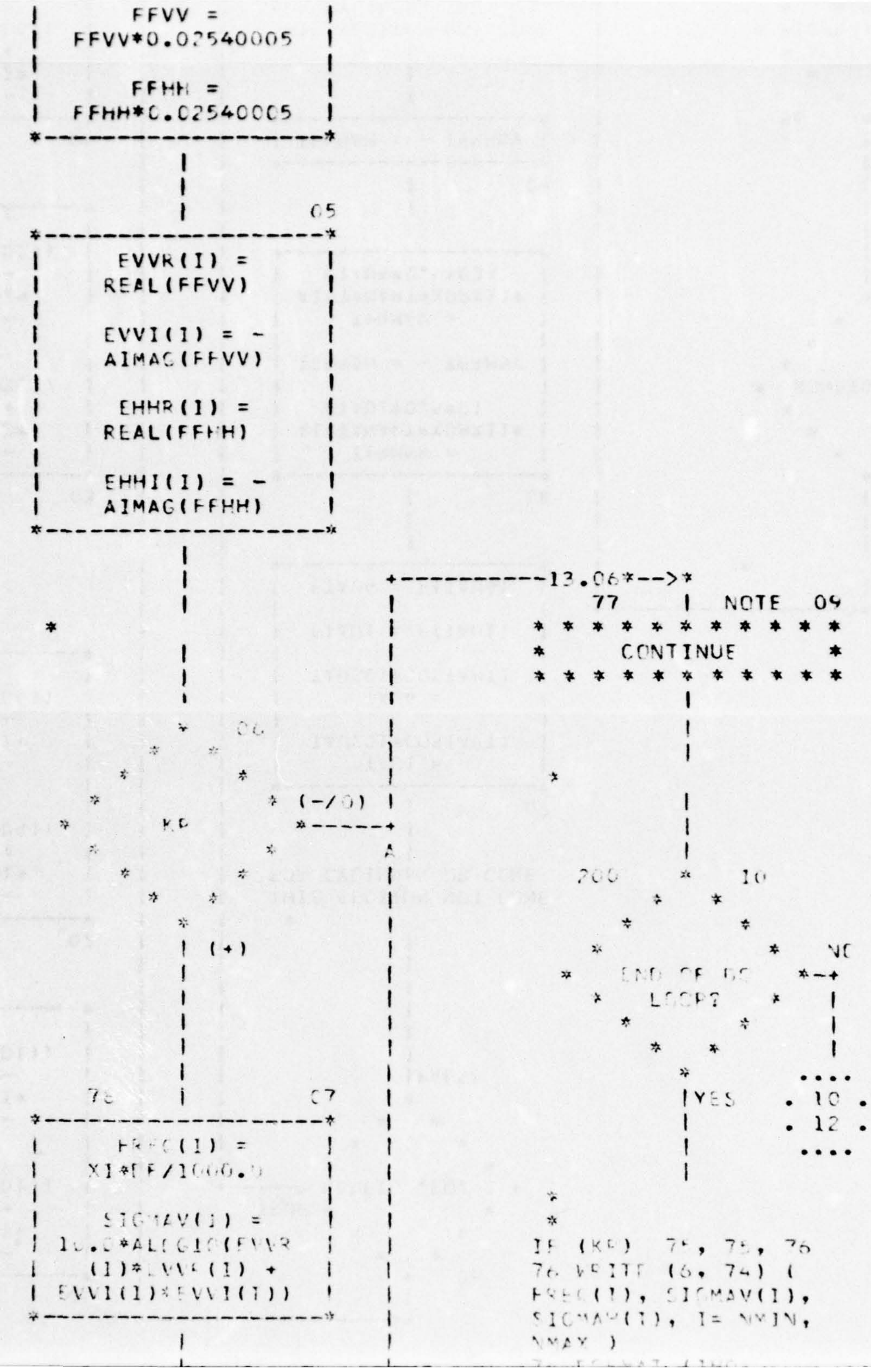

TERM3V = -
A2*((X2MX12*
FSTCOP) +
(XOPX12*SECND3))
*Q3*PHASE
TERM3H = -
A2*((X2MX12*
FSTCOP) -
(XOPX12*SECND3))
*Q3*PHASE
05
KONFIG .EQ. 3
TRUE
FALSE . 13 .
. 01 .
130
THIS SECTION NOT DONE
FOR CYLINDER
DONE FOR CONE
140
10
TAU2 =
TAUSQ2*COS(APT)
TAU3 =
TAUSQ2*COS(AMT)
FTAU2 = F(TAU2)
FTAU3 = F(TAU3)
11
TERM7V = -
A2*TANAPT*XOMX12*
FTAU2*0.5*Q1*
PHASE
TERM7H = - TERM7V
TERM8V = -
A2*TANAMT*XOPX12*
FTAU3*0.5*Q3A*
PHASE
12
TERM8H = - TERM8V
ABS(APT -
PIQVR2) .GT.
WAPT
TRUE
FALSE . 13 .
. 01 .
130
THIS SECTION USED TO
COMPUTE CONE RETURN
NEAR SPECULAR TO
CONIC SURFACE (THET
NEAR (PI/2)-ALPHA)
15
XX =
CMPLX(0.0,1.0)
ZZ =
XOMX12*XX*TAUSQ2*
SIN(APT)/3.0
YY = -
X2PX12*FSTCOP
16
FFVV = -
PISQRT*A2*PHASE*
(YY + ZZ)
FFHH =
PISQRT*A2*PHASE*
(YY - ZZ)
.13.03.
55

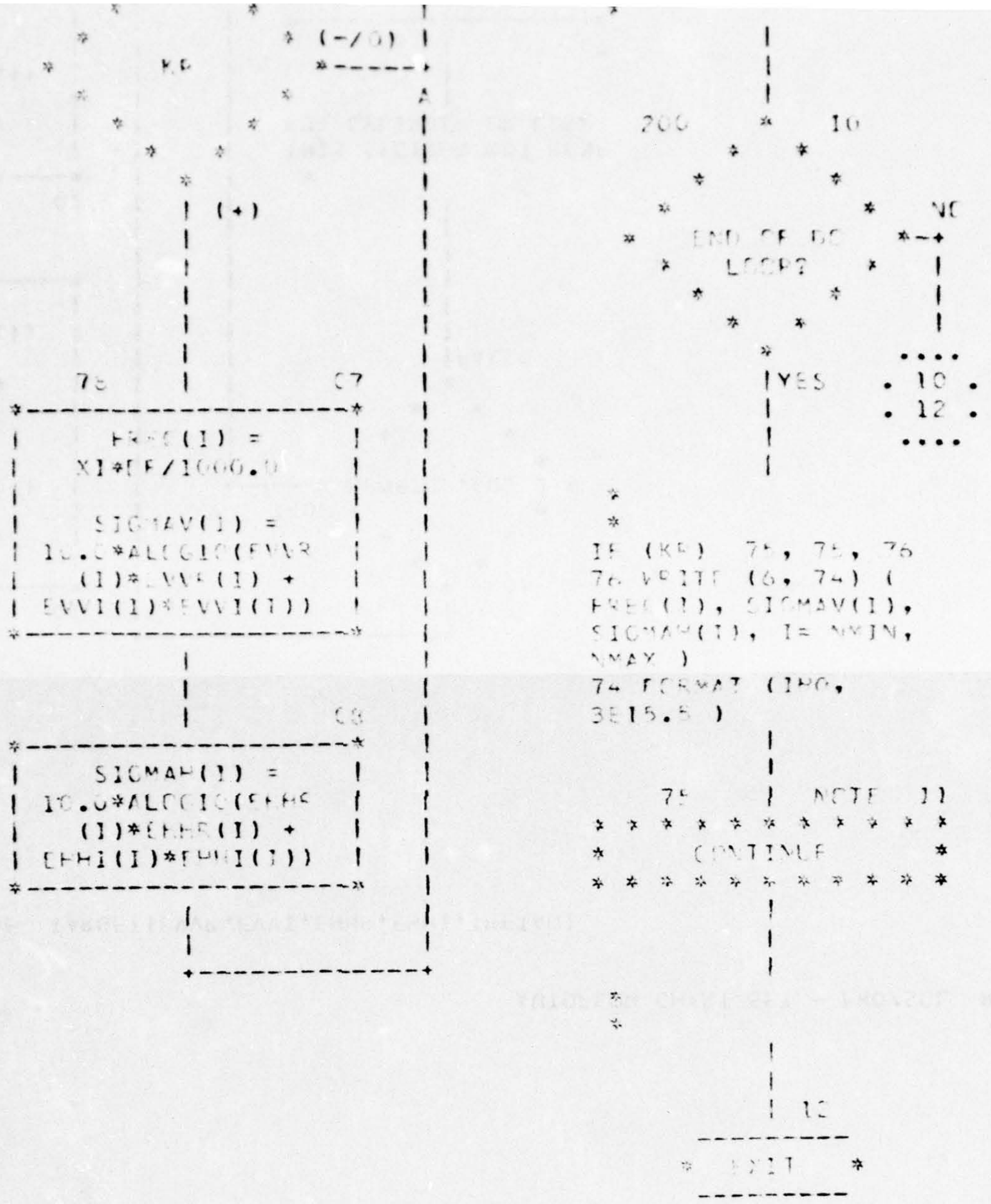
CHART TITLE - SUFROUTINE TARGET(EVVR,EVVI,EHHR,EHHI,THETAD)

L-42

```
          -----------
         /  130     /
         -----------
              |
   12.05*-->|
              |          01
   *------------------------*
   |     FFVV = -          |
   | PISQRT*(TERM1V +      |
   | TERM2V + TERM3V +     |
   | TERM4V + TERM5V +     |
   | TERM6V + TERM7V +     |
   |      TERM8V)          |
   *------------------------*
              |
              |
              |          02
   *------------------------*
   |     FFHH =            |
   | PISQRT*(TERM1H +      |
   | TERM2H + TERM3H +     |
   | TERM4H + TERM5H +     |
   | TERM6H + TERM7H +     |
   |      TERM8H)          |
   *------------------------*
              |
              |
              |
   *
              |
   12.16--->|
         55   |   NOTE  03
   * * * * * * * * * * * *
   *    CONTINUE          *
   * * * * * * * * * * * *
              |
              |
              |          04
   *------------------------*
   |     FFVV =            |
   |  FFVV*0.02540005      |
   |                       |
```

2


FFVV =
FFVV*0.02540005
FFHH =
FFHH*0.02540005
05
EVVR(I) =
REAL(FFVV)
EVVI(I) = -
AIMAG(FFVV)
EHHR(I) =
REAL(FFHH)
EHHI(I) = -
AIMAG(FFHH)
06
KP
(-/0)
(+)
A
13.06
77
NOTE 09
CONTINUE
200
10
END OF DO
LOOP?
NO
YES
10
12
78
07
FREQ(I) =
XI*FF/1000.0
SIGMAV(I) =
10.0*ALOG10(EVVR
(I)*EVVR(I) +
EVVI(I)*EVVI(I))
IF (KP) 75, 75, 76
76 WRITE (6, 74) (
FREQ(I), SIGMAV(I),
SIGMAH(I), I= NMIN,
NMAX)



```
FREQ(I) =
XI*EF/1000.0

SIGMAV(I) =
10.0*ALOG10(EVVR
(I)*EVVR(I) +
EVVI(I)*EVVI(I))

SIGMAH(I) =
10.0*ALOG10(EHHR
(I)*EHHR(I) +
EHHI(I)*EHHI(I))

IF (KP)  75, 75, 76
76 WRITE (6, 74) (
FREQ(I), SIGMAV(I),
SIGMAH(I), I= NMIN,
NMAX )
74 FORMAT (1HO,
3E15.5 )

75 CONTINUE
```

3

CHART TITLE - NON-PROCEDURAL STATEMENTS

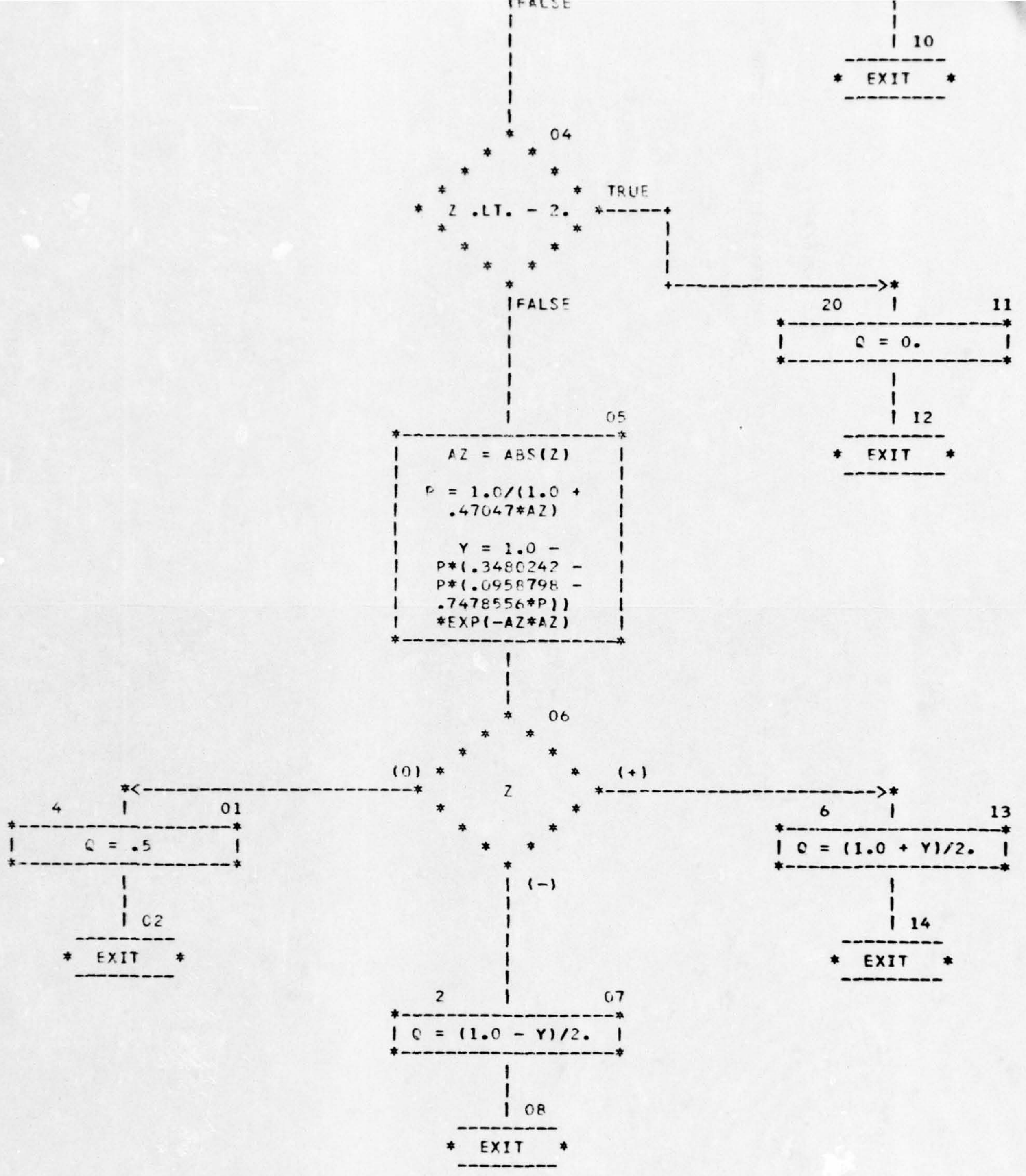
```
         COMMON  MOVER, M, NMIN, NMAX, DF, FC, PW, TO
         DIMENSION  EVVR(512), EVVI(512), EHHR(512), EHHI(512),
                   FREQ(512), SIGMAV(512), SIGMAH(512)
         COMPLEX PHASE, XOPX11, XOMX11, XOPX12, XOMX12, X2PX11, X2MX11,
                 X2PX12, X2MX12, TERM1V, TERM1H, TERM2V, TERM2H, TERM3V,
                 TERM3H, TERM4V, TERM4H, TERM5V, TERM5H, TERM6V, TERM6H,
                 TERM7V, TERM7H, TERM8V, TERM8H, FFVV, FFHH, F, FTAU1,
                 FTAU2, FTAU3, FTAU4      ,XX,YY,ZZ
1000     FORMAT( 3F10.0,2I5 )
3030     FORMAT (1HO///'         FRUSTRA (UFIMTSEV SOLUTION)'//'       FRUSTR
         A LENGTH = ',F10.6,' INCHES'/'    FRONT-END RADIUS = ',F10.6,' INC
         HES'/'     BACK-END RADIUS = ',F10.6,' INCHES'/'  FRUSTRA HALF-ANG
         LE = ',F10.6,' DEGREES')
3040     FORMAT (1HO///'            CONE (UFIMTSEV SOLUTION)'//'          CONE
          LENGTH = ',F10.6,' INCHES'/'            BASE RADIUS = ',F10.6,' INCH
         ES'/'     CONE HALF-ANGLE = ',F10.6,' DEGREES')
3050     FORMAT (1HO///'         CYLINDER (UFIMTSEV SOLUTION)'//'     CYLIND
         ER LENGTH = ',F10.6,' INCHES'/'     CYLINDER RADIUS = ',F10.6,' IN
         CHES')
79       FORMAT(1HO, ' THETA = ' , E15.5  /// )
```

CHART TITLE - FUNCTION FIRS(XN)

```
                  ------------
                 /            /
                 ------------
                       |
                       |
                       |            01
            *----------------------*
            |         PI =         |
            | 3.14159265358979     |
            |                      |
            |         A =          |
            |   (SIN(PI/XN))/XN    |
            |                      |
            |         B =          |
            | 1.0/(COS(PI/XN) -    |
            |        1.0)          |
            |                      |
            |     FIRS = A*B       |
            *----------------------*
                       |
                       |
                       | 02
                  ----------
                *   EXIT   *
                  ----------
```

```
04/26/76                                                 AUTOFLOW CHART SET - FWO/SCL  RADSIM

 CHART TITLE - FUNCTION  SECO(PHI,XN)



                                                 -----------
                                                /           /
                                               -----------
                                                     |
                                                     |
                                                     |          02
                                           *-------------------*
                                           |       PI =        |
                                           | 3.14159265358979  |
                                           |                   |
                                           |        A =        |
                                           |  (SIN(PI/XN))/XN  |
                                           |                   |
                                           | B = COS(PI/XN) -  |
                                           | COS((2.0*PHI)/XN) |
                                           |                   |
                                           |     C = 1.0/B     |
                                           *-------------------*
                                                     |
                                                     |
                                                     |          03
                                           *-------------------*
                                           |    SECO = A*C     |
                                           *-------------------*
                                                     |
                                                     |
                                                     | 04
                                                 ----------
                                               *   EXIT    *
                                                 ----------
```

```
[illegible]
                                                   AUTOFLOW CHART SET - FWD/SCL  RADSIM
CHART TITLE - FUNCTION  Q(Z)
L-46



                                         Q(Z) = 0.5*(1 +
                                         ERF(Z))
                                          * ERF(Z) IS
                                         EVALUATED USING A
                                         RATIONAL POLYNOMIAL
                                         APPROXIMATION
                                          * REFERENCE (HANDBK
                                         MATH FUNCT BY
                                         ABRAMOWITZ AND
                                         STEGUN,
                                          *          SECTION
                                         7.1.26)

                                               *
                                               I
                                               *  03
                                             *   *
                                           *       *
                                         *           *  TRUE
                                       *  Z .GT. 2.   *----------------------->*
                                         *           *                  10    I            09
                                           *       *                  *-----------------------*
                                             *   *                    I        Q = 1.         I
                                               *                      *-----------------------*
                                               IFALSE                             I
                                               I                                  I
                                               I                                  I 10
                                               I                              ----------
                                               I                             *  EXIT    *
                                               I                              ----------
                                               I
                                               *  04
                                             *   *
                                           *       *
                                         *           *  TRUE
                                       *  Z .LT. - 2.  *-----+
                                         *           *       I
                                           *       *         I
                                             *   *           +----------------->*
                                               *                        20    I            11
                                               IFALSE                 *-----------------------*
                                               I                      I        Q = 0.         I
                                               I                      *-----------------------*
```

FALSE
10
EXIT
04
Z .LT. - 2.
TRUE
FALSE
20
11
Q = 0.
12
EXIT
05
AZ = ABS(Z)
P = 1.0/(1.0 +
.47047*AZ)
Y = 1.0 -
P*(.3480242 -
P*(.0958798 -
.7478556*P))
*EXP(-AZ*AZ)
06
Z
(0)
(+)
(-)
4
01
Q = .5
02
EXIT
6
13
Q = (1.0 + Y)/2.
14
EXIT
2
07
Q = (1.0 - Y)/2.
08
EXIT

2

```
  04/26/76                                                 AUTOFLOW CHART SET - FWO/SCL  RADSIM

   CHART TITLE - SUBROUTINE  BESL(X,B0,B1,B2)


         ------------
         /   BESL   /
         -----------

              *

     * BESSEL FUNCTION
    SUBROUTINE UTILIZING
    POLYNOMIAL
    APPROXIMATIONS
     * COMPUTES J0,J1,OR
    J2 FOR POSITIVE REAL
    ARGUMENTS
     * REFERENCE (HNDBK
    MATH FUNCT BY
    ABRAMOWITZ AND STEGUN
    SECTION 9.4 )
              *
              |        02
    *-------------------*
    |     S = 1.0       |
    *-------------------*
              |
              |
              |
              *   03                                          +-------------------->*
            *   *                                             |              9     |        13
          *       *                                           |          *-------------------*
  FALSE *           *                                         |          |     X2 = 3./X     |
+-----*  X .LT. 0.0   *                                       |          |                   |
|       *           *                                         |          | F0 = .79788456 +  |
|         *       *                                           |          |   X2*(-.77E-6 +   |
|           *   *                                             |          | X2*(-.00552740 +  |
|             *                                               |          |  X2*(-.9512E-4 +  |
|             |TRUE                                           |          |  Y2*(.00137237 +  |
|             |                                               |          | X2*(-.72805E-3 +  |
|             |                                               |          | X2*0.14476E-3)))) |
|             |                                               |          |         )         |
|             |                                               |          *-------------------*
|             |                                               |                    |
|             |        04                                     |                    |
|   *-------------------*                                     |                    |        14
|   |    S = - 1.0      |                                     |          *-------------------*
|   *-------------------*                                     |          |    T0 = X -       |
|             |                                               |          |   .78539816 +     |
+------------>|                                               |          | X2*(-.04166397 +  |
              |        05                                     |          |  X2*(-.3954E-4 +  |
    *-------------------*                                     |          |  X2*(.00262573 +  |
    |    X = ABS(X)     |                                     |          | X2*(-.00054125 +  |
    *-------------------*                                     |          | X2*(-.00029333 +  |
              |                                               |          | X2*0.00013558)))) |
```

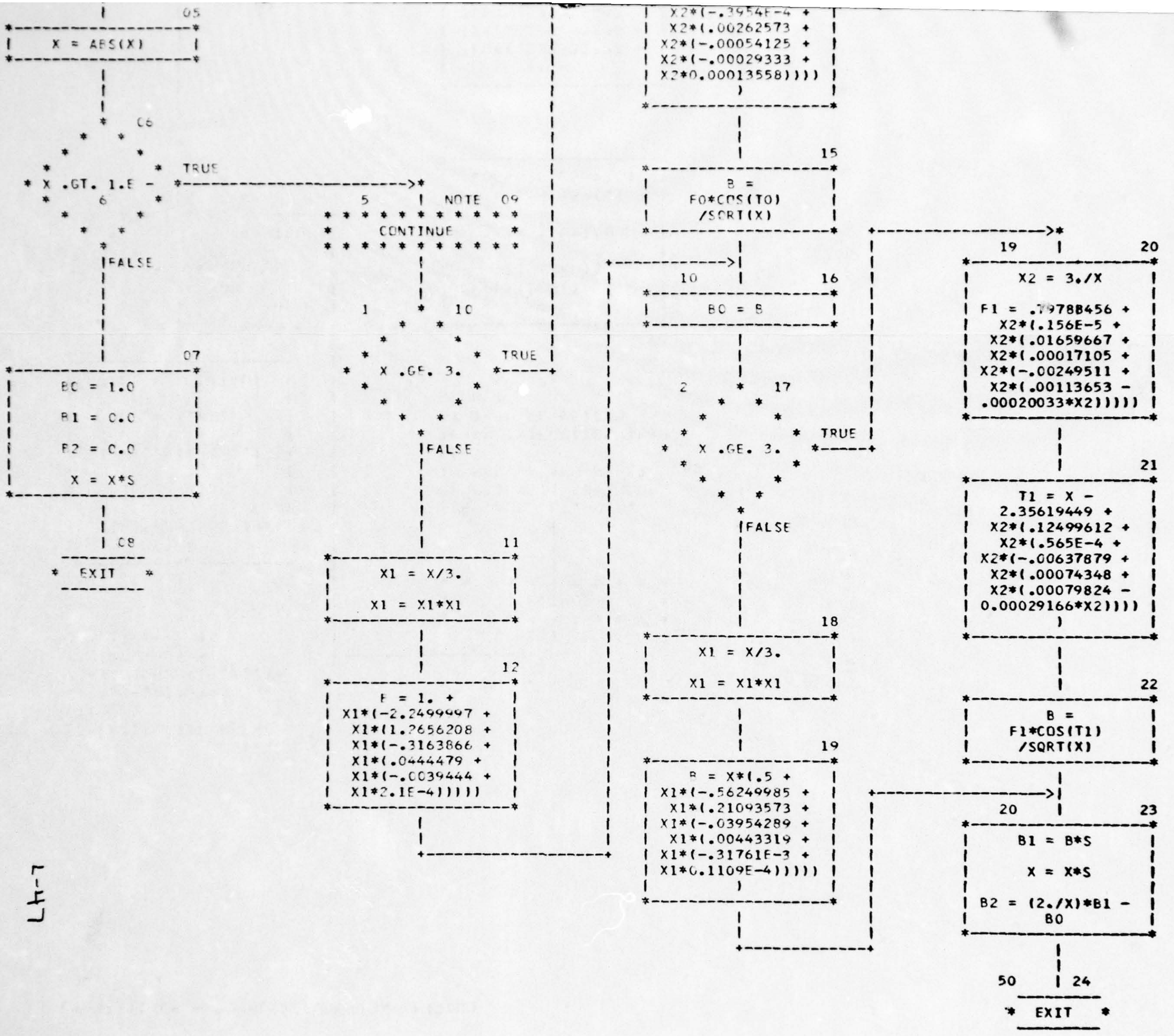
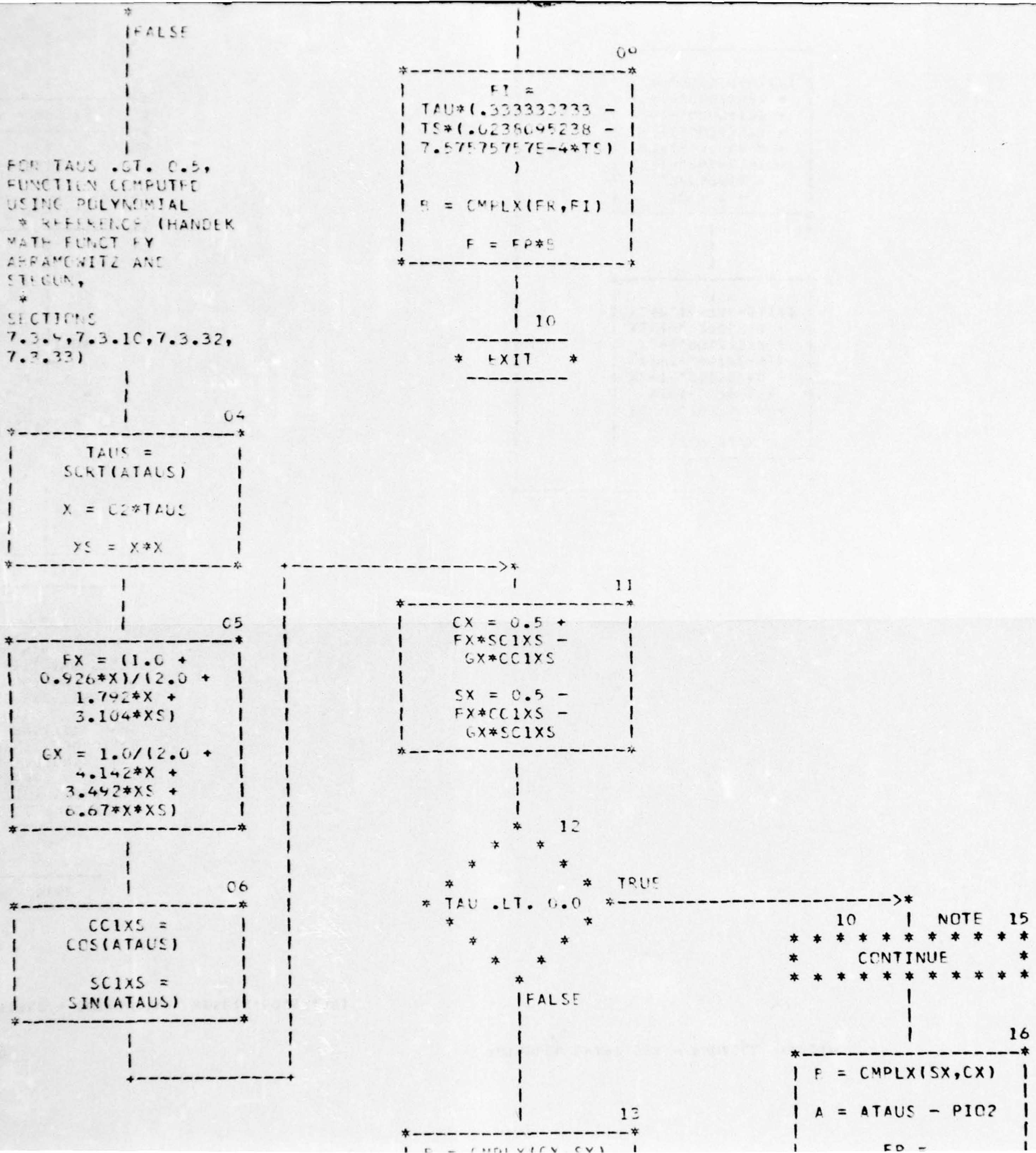

05
X = ABS(X)
06
X .GT. 1.E - 6
TRUE
FALSE
07
BC = 1.0
B1 = C.C
B2 = C.0
X = X*S
C8
EXIT
5
NOTE 09
CONTINUE
1
10
X .GE. 3.
TRUE
FALSE
11
X1 = X/3.
X1 = X1*X1
12
F = 1. +
X1*(-2.2499997 +
X1*(1.2656208 +
X1*(-.3163866 +
X1*(.0444479 +
X1*(-.0039444 +
X1*2.1E-4)))))
X2*(-.3954E-4 +
X2*(.00262573 +
X2*(-.00054125 +
X2*(-.00029333 +
X2*0.00013558))))
)
15
B =
F0*COS(T0)
/SQRT(X)
10
16
B0 = B
2
17
X .GE. 3.
TRUE
FALSE
18
X1 = X/3.
X1 = X1*X1
19
B = X*(.5 +
X1*(-.56249985 +
X1*(.21093573 +
X1*(-.03954289 +
X1*(.00443319 +
X1*(-.31761E-3 +
X1*0.1109E-4))))))
)
19
20
X2 = 3./X
F1 = .79788456 +
X2*(.156E-5 +
X2*(.01659667 +
X2*(.00017105 +
X2*(-.00249511 +
X2*(.00113653 -
.00020033*X2)))))
21
T1 = X -
2.35619449 +
X2*(.12499612 +
X2*(.565E-4 +
X2*(-.00637879 +
X2*(.00074348 +
X2*(.00079824 -
0.00029166*X2))))
)
22
B =
F1*COS(T1)
/SQRT(X)
20
23
B1 = B*S
X = X*S
B2 = (2./X)*B1 -
B0
50
24
EXIT

CHART TITLE - COMPLEX FUNCTION F(TAU)

L- 48

```
COMPUTES FTAU WHERE
FTAU
=(EXP(-J*TAU**2)
/2*TAU)*SQRT(PI/2.)*
(C2(TAU**2) +
J*S2(TAU**2))                +------------------->*
                             |           20       |  NOTE  07
                             |          * * * * * * * * * * *
         *                   |          *      CONTINUE       *
         |          02       |          * * * * * * * * * * *
*--------------------*       |                    |
|       PI =         |       |                    |
| 3.14159265358979   |       |                    |
|                    |       |          FOR TAUS .LE. 0.5,
|   PIO2 = PI/2.     |       |          FUNCTION IS EXPANDED
|                    |       |          IN SERIES AND FIRST
| C1 = SQRT(PI/2.)   |       |          FEW
|                    |       |          TERMS INTEGRATED TERM
|    C2 = 1./C1      |       |          BY TERM TO OBTAIN
|                    |       |          RESULT
| ATAUS = ABS(TAU)   |       |                    |
*--------------------*       |                    |
         |                   |                    |              08
         |                   |          *----------------------*
         |                   |          |         FR =         |
         *     03            |          | CMPLX(COS(TAU), -    |
      *     *                |          |    SIN(TAU))         |
    *         *              |          |                      |
  *             *   TRUE     |          |   TS = TAU*TAU       |
 *  ATAUS .LE.    *----------+          |                      |
  *     0.5     *                       | FR = 1 - TS*(.1 -    |
    *         *                         |  .0046296296*TS)     |
      *     *                           *----------------------*
         *                                        |
         |FALSE                                   |
         |                                        |              09
         |                              *----------------------*
```

2



```
        *                                         |
        |FALSE                                    |
        |                                         |            09
        |                                *--------------------*
        |                                |        FI =        |
        |                                | TAU*(.333333333 -  |
        |                                | TS*(.0238095238 -  |
        |                                | 7.57575757E-4*TS)  |
FOR TAUS .GT. 0.5,                       |         )          |
FUNCTION COMPUTED                        |                    |
USING POLYNOMIAL                         | B = CMPLX(FR,FI)   |
 * REFERENCE (HANDBK                     |                    |
MATH FUNCT BY                            |     F = FP*B       |
ABRAMOWITZ AND                           *--------------------*
STEGUN,                                           |
 *                                                |
SECTIONS                                          | 10
7.3.9,7.3.10,7.3.32,                          ---------
7.3.33)                                       *  EXIT   *
        |                                     ---------
        |
        |            04
*--------------------*
|       TAUS =       |
|   SQRT(ATAUS)      |
|                    |
|    X = C2*TAUS     |
|                    |
|     XS = X*X       |
*--------------------*   *-------------------->*
        |                |                     |             11
        |            05  |            *--------------------*
*--------------------*   |            |     CX = 0.5 +     |
|    FX = (1.0 +     |   |            |    FX*SC1XS -      |
| 0.926*X)/(2.0 +    |   |            |     GX*CC1XS       |
|      1.792*X +     |   |            |                    |
|     3.104*XS)      |   |            |     SX = 0.5 -     |
|                    |   |            |    FX*CC1XS -      |
|   GX = 1.0/(2.0 +  |   |            |     GX*SC1XS       |
|      4.142*X +     |   |            *--------------------*
|     3.492*XS +     |   |                     |
|     6.67*X*XS)     |   |                     |
*--------------------*   |                     *      12
        |                |                  *     *
        |                |               *           *
        |            06  |            *                 *  TRUE
*--------------------*   |         * TAU .LT. 0.0          *------------------->*
|      CC1XS =       |   |            *                 *            10   | NOTE  15
|    COS(ATAUS)      |   |               *           *           * * * * * * * * * * *
|                    |   |                  *     *              *      CONTINUE       *
|      SC1XS =       |   |                     *                 * * * * * * * * * * *
|    SIN(ATAUS)      |   |                     |FALSE                     |
*--------------------*   |                     |                          |             16
        |                |                     |                 *--------------------*
        *----------------*                     |                 | F = CMPLX(SX,CX)   |
                                               |             13  |                    |
                                      *--------------------*     | A = ATAUS - PIO2   |
                                      | F = CMPLX(CX,SX)   |     |                    |
                                                                 |       FP =         |
```

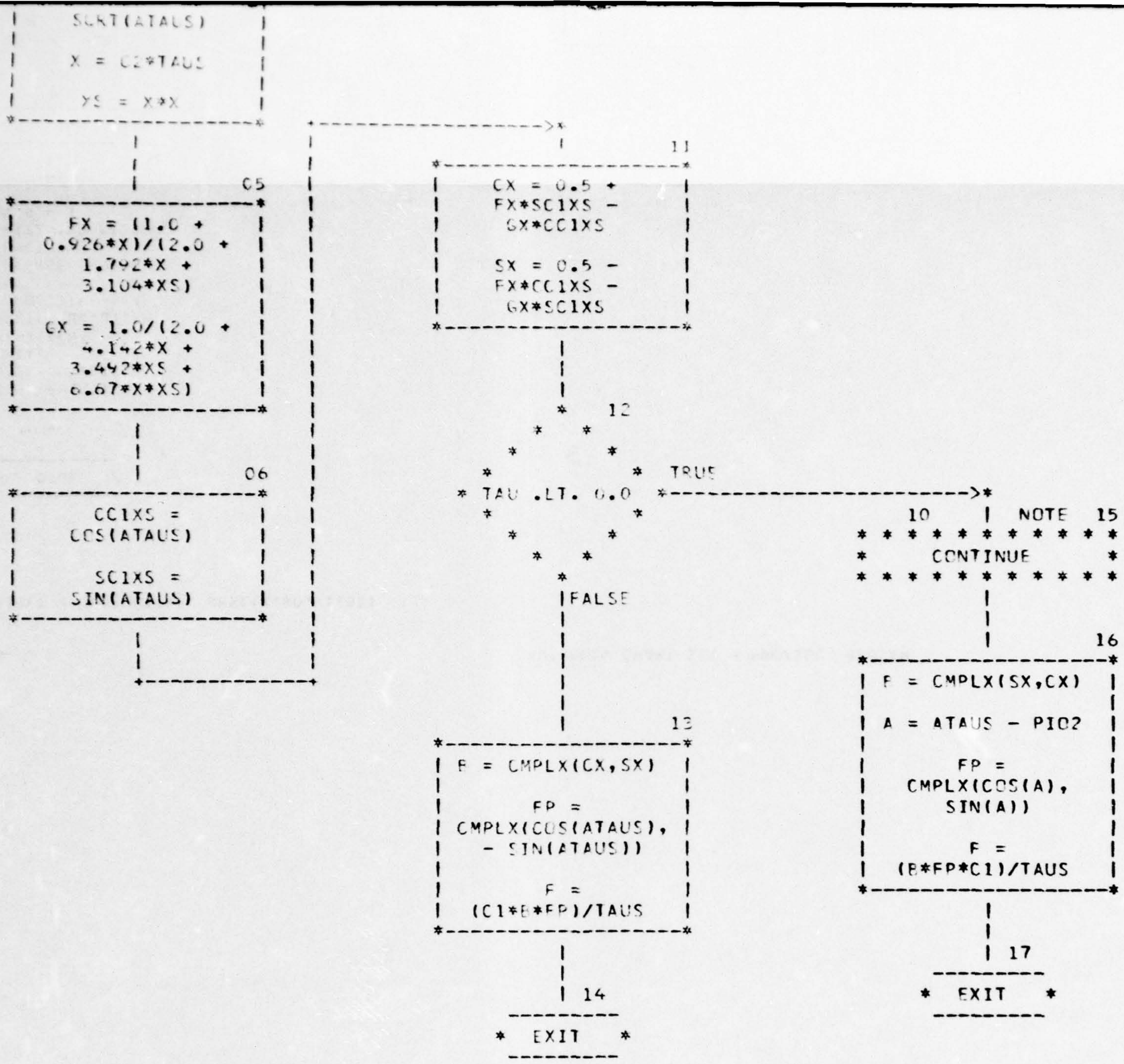

SQRT(ATAUS)
X = C2*TAUS
XS = X*X
05
FX = (1.0 + 0.926*X)/(2.0 + 1.792*X + 3.104*XS)
GX = 1.0/(2.0 + 4.142*X + 3.492*XS + 6.67*X*XS)
06
CC1XS = COS(ATAUS)
SC1XS = SIN(ATAUS)
11
CX = 0.5 + FX*SC1XS - GX*CC1XS
SX = 0.5 - FX*CC1XS - GX*SC1XS
12
TAU .LT. 0.0
TRUE
FALSE
10
NOTE 15
CONTINUE
13
B = CMPLX(CX,SX)
FP = CMPLX(COS(ATAUS), - SIN(ATAUS))
F = (C1*B*FP)/TAUS
14
EXIT
16
B = CMPLX(SX,CX)
A = ATAUS - PIO2
FP = CMPLX(COS(A), SIN(A))
F = (B*FP*C1)/TAUS
17
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
                COMPLEX E, EP
          SUBROUTINE TARGET ( EVVR, EVVI, EHHR, EHHI, THETAT)                  RCS5 001
      C                                                                      * RCS5 002
      C * * * * * GENERALIZED PROGRAM FOR A FRUSTRA, CONE, OR CYLINDER  * * * RCS5 003
      C                (UFIMTSEV SOLUTION FOR CW)                            * RCS5 004
      C                                                                      * RCS5 005
      C          A1     = FRONT-END RADIUS (INCHES)                          * RCS5 006
      C          A2     = BACK-END RADIUS (SHOULD BE .GE. A1) - (INCHES)     * RCS5 007
      C          H2     = TOTAL LENGTH (INCHES)                              * RCS5 008
      C          ALPHA  = CONE OR FRUSTRA HALF ANGLE (DEGREES)               * RCS5 009
      C          FREQ   = CARRIER FREQUENCY (GHZ)                            * RCS5 010
      C          DELTHT = ASPECT ANGLE INCREMENT (DEGREES) - (.GE. 0.1)      * RCS5 011
      C          KONFIG = TARGET CONFIGURATION                               * RCS5 012
      C                     1 = FRUSTRA                                      * RCS5 013
      C                     2 = CONE                                         * RCS5 014
      C                     3 = CYLINDER                                     * RCS5 015
      C                                                                      * RCS5 016
      C * * * * * ALL DIMENSIONS ARE IN INCHES AND ANGLES ARE IN DEGREES * * * RCS5 017
```

```
C                                                                     * RCS5 016
C * * * * ALL DIMENSIONS ARE IN INCHES AND ANGLES ARE IN DECREES  * * * RCS5 017
C                                                                     * RCS5 018
      COMMON  MOVER, M, NMIN, NMAX, DF, FC, PW, TO                      RCS5 019
C     NMIN = MINIMUM FREQUENCY SAMPLE                                   RCS5 020
C     NMAX = MAXIMUM FREQUENCY SAMPLE                                   RCS5 021
C     DF   = FREQUENCY INCREMENT IN MHZ                                 RCS5 022
C     FC   = CARRIER FREQUENCY IN GHZ                                   RCS5 023
      DIMENSION  FVVR(512), FVVI(512), FHHR(512), FHHI(512),            RCS5 024
     1           FREQ(512), SIGMAV(512), SIGMAH(512)                    RCS5 025
C                                                                       RCS5 026
      COMPLEX PHASE, XOPX11, XOMX11, XOPX12, XOMX12, X2PX11, X2MX11,    RCS5 027
     1       X2PX12, X2MX12, TERM1V, TERM1H, TERM2V, TERM2H, TERM3V,    RCS5 028
     2       TERM3H, TERM4V, TERM4H, TERM5V, TERM5H, TERM6V, TERM6H,    RCS5 029
     3       TERM7V, TERM7H, TERM8V, TERM8H, FFVV, FFHH, F, FTAU1,      RCS5 030
     +       FTAU2, FTAU3, FTAU4       ,XX,YY,ZZ                        RCS5 031
C                                                                     * RCS5 032
C                                                                     * RCS5 033
C                                                                     * RCS5 034
      READ(5,1000) A1,A2,H2, KONFIG, KP                                 RCS5 035
 1000 FORMAT( 3F10.0,2I5 )                                              RCS5 036
      ALF    = A2 - A1                                                  RCS5 037
      ALPHA  = ATAN2(ALF, H2)                                           RCS5 038
      PI     = 3.14159265358979                                         RCS5 039
      ANGLE  = (ALPHA * 180.0) / PI                                     RCS5 040
C                                                                     * RCS5 041
      IF (KONFIG - 2) 30, 40, 50                                        RCS5 042
     +        FTAU2, FTAU3, FTAU4        ,XX,YY,ZZ                         RCS5 031
C                                                                        * RCS5 032
C                                                                        * RCS5 033
C                                                                        * RCS5 034
          READ(5,1000) A1,A2,H2, KONFIG, KP                                RCS5 035
 1000 FORMAT( 3F10.0,2I5 )                                                 RCS5 036
          ALP      = A2 - A1                                               RCS5 037
          ALPHA    = ATAN2(ALP, H2)                                        RCS5 038
          PI       = 3.14159265358979                                      RCS5 039
          ANGLE    = (ALPHA * 180.0) / PI                                  RCS5 040
C                                                                        * RCS5 041
          IF (KONFIG - 3) 30, 40, 50                                       RCS5 042
C                                                                        * RCS5 043
    30 WRITE (6, 3030) H2, A1, A2, ANGLE                                   RCS5 044
 3030 FORMAT (1HO///'          FRUSTRA (OPTIMALLY SOLUTION)'//'      FRUSTRCS5 045
       1A LENGTH = ',F10.6,' INCHES'/'     FRONT-END RADIUS = ',F10.6,' INCRCS5 046
       2HES'/'       BACK-END RADIUS = ',F10.6,' INCHES'/'  FRUSTRA HALF-ANGRCS5 047
       3LE = ',F10.6,' DEGREES')                                           RCS5 048
          GO TO 60                                                         RCS5 049
C                                                                        * RCS5 050
    40 WRITE (6, 3040) H2, A2, ANGLE                                       RCS5 051
 3040 FORMAT (1HO///'           CONE (OPTIMALLY SOLUTION)'//'         CONERCS5 052
       1 LENGTH = ',F10.6,' INCHES'/'           BASE RADIUS = ',F10.6,' INCHRCS5 053
       2ES'/'      CONE HALF-ANGLE = ',F10.6,' DEGREES')                   RCS5 054
          GO TO 60                                                         RCS5 055
```

3

L-49a

```
CARD NO       ****                          CONTENTS                                   ****

   348        C                                                                  * RCS5 056
   349           50 WRITE (6, 3050) H2, A2                                         RCS5 057
   350         3050 FORMAT (1H0///'        CYLINDER (UFIMTSEV SOLUTION)'//'    CYLINDRCS5 058
   351             1ER LENGTH = ',F10.6,' INCHES'/'     CYLINDER RADIUS = ',F10.6,' INRCS5 059
   352             2CHES')                                                         RCS5 060
   353        C                                                                  * RCS5 061
   354           60 WRITE (6, 79) THETAD                                           RCS5 062
   355           79 FORMAT(1H0, ' THETA = ' , F15.5  /// )                         RCS5 063
   356        C                                                                  * RCS5 064
   357              C       = 11.80285078                                          RCS5 065
   358              DTR     = PI / 180.0                                           RCS5 066
   359              PISQRT = SQRT(PI)                                              RCS5 067
   360              PIOVR2 = PI / 2.0                                              RCS5 068
   361              A12       = 2.0* A1                                            RCS5 069
   362              A22       = 2.0* A2                                            RCS5 070
   363              H22       = 2.0* H2                                            RCS5 071
   364              XNPOS   = (3.0 / 2.0) + (ALPHA / PI)                           RCS5 072
   365              XNNEG   = (3.0 / 2.0) - (ALPHA / PI)                           RCS5 073
   366        C     COMPUTE C(N)                                                   RCS5 074
   367              FSTCOP = FIRS(XNPOS)                                           RCS5 075
   368              FSTCON = FIRS(XNNEG)                                           RCS5 076
   369              THETA     = THETAD * DTR                                       RCS5 077
   370              STHT      = SIN(THETA)                                         RCS5 078
```

2

```
      THETA   = THETAD * DTR                                    RCS5 077
      STHT    = SIN(THETA)                                      RCS5 078
      CTHT    = COS(THETA)                                      RCS5 079
      APT     = ALPHA + THETA                                   RCS5 080
      AMT     = ALPHA - THETA                                   RCS5 081
      TANAPT = TAN(APT)                                         RCS5 082
      TANAMT = TAN(AMT)                                         RCS5 083
      TWOPIC     = 2.0 * PI / C                                 RCS5 084
C                                                             * RCS5 085
      DO 200  I = NMIN,NMAX                                     RCS5 086
      XI      = I - 1                                           RCS5 087
      XKO     = TWOPIC * XI * DF / 1000.0                       RCS5 088
      X2KA1     = A12 * XKO                                     RCS5 089
      X2KA2     = A22 * XKO                                     RCS5 090
      X2KH2     = H22 * XKO                                     RCS5 091
C                                                             * RCS5 092
      IF (KONFIG .EQ. 3) GO TO 70                               RCS5 093
      TAUSQ1 = X2KA1 / SIN(ALPHA)                               RCS5 094
      TAUSQ2 = X2KA2 / SIN(ALPHA)                               RCS5 095
   70 CONTINUE                                                  RCS5 096
C                                                             * RCS5 097
      RAPT    = PIOVR2 - ACOS(0.8 * COS(ALPHA) / X2KH2)         RCS5 098
      ARG1    = X2KA1 * STHT                                    RCS5 099
      ARG2    = X2KA2 * STHT                                    RCS5 100
      FASE    = X2KH2 * CTHT                                    RCS5 101
      PHASE   = CMPLX (COS(FASE), SIN(FASE))                    RCS5 102
C                                                             * RCS5 103
      CALL BESL (ARG1, XJ01, XJ11, XJ21)                        RCS5 104
      C                                                              * RCS5 097
                KAPT   = PIOVR2 - ACOS(0.8 * COS(ALPHA) / X2KH2)       RCS5 098
                ARG1   = X2KA1 * STHT                                  RCS5 099
                ARG2   = X2KA2 * STHT                                  RCS5 100
                FASE   = X2KH2 * CTHT                                  RCS5 101
                PHASE  = CMPLX (COS(FASE), SIN(FASE))                  RCS5 102
      C                                                              * RCS5 103
                CALL BESL (ARG1, XJ01, XJ11, XJ21)                     RCS5 104
                CALL BESL (ARG2, XJ02, XJ12, XJ22)                     RCS5 105
      C                                                              * RCS5 106
                X0PX11 = CMPLX(XJ01, XJ11)                             RCS5 107
                X0MX11 = CMPLX(XJ01,-XJ11)                             RCS5 108
                X0PX12 = CMPLX(XJ02, XJ12)                             RCS5 109
                X0MX12 = CMPLX(XJ02,-XJ12)                             RCS5 110
                X2PX11 = CMPLX(XJ21, XJ11)                             RCS5 111
                X2MX11 = CMPLX(XJ21,-XJ11)                             RCS5 112
                X2PX12 = CMPLX(XJ22, XJ12)                             RCS5 113
```

3

```
C4/26/76        INPUT LISTING                          AUTOFLOW CHART SET - FWD/SCL  RADSIM

  CARD NO          ****                       CONTENTS                                    ****

    406                  X2MX12 = CMPLX(XJ22,-XJ12)                                    RCS5 114
    407            C                                                                 * RCS5 115
    408                  Q1     = Q(X2KA1 * (PI - APT))                                RCS5 116
    409                  Q4     = Q(X2KA1 * (PIOVR2 - THETA))                          RCS5 117
    410                  Q3     = Q(X2KA2 * ((THETA+THETA-PI) * (-AMT)) / (PIOVR2-ALPHA))  RCS5 118
    411                  Q3A    = Q(X2KA2 * AMT)                                       RCS5 119
    412            C                                                                 * RCS5 120
    413                  IF (THETA .GT. PIOVR2)  GO TO 90                              RCS5 121
    414            C     DEFINE ANGLES FOR THET .LT. 90 DEGREES                        RCS5 122
    415            C                                                                 * RCS5 123
    416                  PHI1   = PIOVR2 + THETA                                       RCS5 124
    417                  PHI4   = PIOVR2 - THETA                                       RCS5 125
    418                  PHI2   = APT                                                  RCS5 126
    419                  GO TO 98                                                      RCS5 127
    420            C     DEFINE ANGLES FOR THET .GT. 90 DEGREES                        RCS5 128
    421               90 PHI1   = PI-APT                                               RCS5 129
    422                  PHI4   = 0.0                                                  RCS5 130
    423                  PHI2   = (3.0 * PIOVR2) - THETA                               RCS5 131
    424            C                                                                 * RCS5 132
    425               98 IF ( Q3A .EQ. 0.0 ) GO TO 99                                  RCS5 133
    426                  PHI3 = AMT                                                    RCS5 134
```

AD-A031 440   GENERAL DYNAMICS FORT WORTH TEX CONVAIR AEROSPACE DIV   F/G 17/9
ENDO ATMOSPHERIC EXO ATMOSPHERIC RADAR MODELING. RADAR CROSS SE--ETC(U)
JUN 76   R J HANCOCK, F H CLEVELAND   F30602-73-C-0380

UNCLASSIFIED   RADC-TR-76-186-VOL-4-PT-2   NL

2 OF 4
AD
A031440

```
      PHI2   = (3.0 * PIDVR2) - THETA                                RCS5 131
C                                                                  * RCS5 132
   98 IF ( 63A .EQ. 0.0 ) GO TO 99                                   RCS5 133
      PHI3 = AMT                                                     RCS5 134
      GL TC 100                                                      RCS5 135
   99 PHI3 = PIDVR2 - THETA                                          RCS5 136
C                                                                  * RCS5 137
C     COMPUTE B(N,PHI)                                               RCS5 138
  100 SECND1 = SECO(PHI1, XNNEG)                                     RCS5 139
      SECND4 = SECO(PHI4, XNNEG)                                     RCS5 140
      SECND2 = SECO(PHI2, XNPOS)                                     RCS5 141
      SECND3 = SECO(PHI3, XNPOS)                                     RCS5 142
C                                                                  * RCS5 143
      TERM1V = 0.0                                                   RCS5 144
      TERM1H = 0.0                                                   RCS5 145
      TERM4V = 0.0                                                   RCS5 146
      TERM4H = 0.0                                                   RCS5 147
      TERM5V = 0.0                                                   RCS5 148
      TERM5H = 0.0                                                   RCS5 149
      TERM6V = 0.0                                                   RCS5 150
      TERM6H = 0.0                                                   RCS5 151
      TERM7V = 0.0                                                   RCS5 152
      TERM7H = 0.0                                                   RCS5 153
      TERM8V = 0.0                                                   RCS5 154
      TERM8H = 0.0                                                   RCS5 155
C                                                                  * RCS5 156
      IF (KONFIG .EQ. 2) GO TO 120                                   RCS5 157
C                                                                  * RCS5 158
```

2

```
      TERM4V = 0.0                                                      RCS5 146
      TERM4H = 0.0                                                      RCS5 147
      TERM5V = 0.0                                                      RCS5 148
      TERM5H = 0.0                                                      RCS5 149
      TERM6V = 0.0                                                      RCS5 150
      TERM6H = 0.0                                                      RCS5 151
      TERM7V = 0.0                                                      RCS5 152
      TERM7H = 0.0                                                      RCS5 153
      TERM8V = 0.0                                                      RCS5 154
      TERM8H = 0.0                                                      RCS5 155
C                                                                     * RCS5 156
      IF (KONFIG .EQ. 2) GO TO 120                                      RCS5 157
C                                                                     * RCS5 158
      TERM1V = -A1 * ((X2PX11 * FSTCON) + (XOMX11 * SECND1)) * Q1       RCS5 159
      TERM1H = -A1 * ((X2PX11 * FSTCON) - (XOMX11 * SFCND1)) * Q1       RCS5 160
      TERM4V = -A1 * ((X2MX11 * FSTCON) + (XOPX11 * SECND4)) * Q4       RCS5 161
      TERM4H = -A1 * ((X2MX11 * FSTCON) - (XOPX11 * SECND4)) * Q4       RCS5 162
C                                                                     * RCS5 163
  120 TERM2V = -A2 * ((X2PX12 * FSTCOP) + (XOMX12 * SECND2)) * PHASE    RCS5 164
      TERM2H = -A2 * ((X2PX12 * FSTCOP) - (XOMX12 * SFCND2)) * PHASF    RCS5 165
      TERM3V = -A2 * ((X2MX12 * FSTCOP) + (XOPX12 * SECND3)) * Q3*PHASE RCS5 166
      TERM3H = -A2 * ((X2MX12 * FSTCOP) - (XOPX12 * SECND3)) * Q3*PHASE RCS5 167
C                                                                     * RCS5 168
      IF (KONFIG .EQ. 3) GO TO 130                                      RCS5 169
      IF (KONFIG .EQ. 2) GO TO 140                                      RCS5 170
C                                                                     * RCS5 171
```

3

L-496

L-49C

```
CARD NO     ****                        CONTENTS                            ****

  464       C     THIS SECTION NOT DONE FOR CYLINDER OR CONE                RCS5 172
                  TAU1   = TAUSQ1 * COS(APT)                                RCS5 173
  465
  466             TAU4   = TAUSQ1 * COS(AMT)                                RCS5 174
  467             FTAU1  = F(TAU1)                                          RCS5 175
  468             FTAU4  = F(TAU4)                                          RCS5 176
  469             TERM5V = A1 * TANAPT * XOMX11 * FTAU1 * 0.5 * Q1          RCS5 177
  470             TERM5H = -TERM5V                                          RCS5 178
  471             TERM6V = A1 * TANAMT * XOPX11 * FTAU4 * 0.5 * Q3A         RCS5 179
  472             TERM6H = -TERM6V                                          RCS5 180
  473       C     THIS SECTION NOT DONE FOR CYLINDER                        RCS5 181
  474       C                DONE FOR CONE                                  RCS5 182
  475           140 TAU2   = TAUSQ2 * COS(APT)                              RCS5 183
  476             TAU3   = TAUSQ2 * COS(AMT)                                RCS5 184
  477             FTAU2  = F(TAU2)                                          RCS5 185
  478             FTAU3  = F(TAU3)                                          RCS5 186
  479             TERM7V =-A2 * TANAPT * XOMX12 * FTAU2 * 0.5 * Q1 * PHASE  RCS5 187
  480             TERM7H = -TERM7V                                          RCS5 188
  481             TERM8V =-A2 * TANAMT * XOPX12 * FTAU3 * 0.5 * Q3A* PHASE  RCS5 189
  482             TERM8H = -TERM8V                                          RCS5 190
  483       C                                                             * RCS5 191
  484             IF (KONFIG - 2) 130, 56, 130                              RCS5 192
  485          56 IF (ABS(APT-PIOVR2) .GT. WAPT ) GO TO 130                 RCS5 193
```

2

```
      TERM8H = -TERM8V                                                   RCS5 190
C                                                                       * RCS5 191
      IF (KONFIG - 2) 130, 56, 130                                        RCS5 192
   56 IF (ABS(APT-PIOVR2) .GT. WAPT ) GO TO 130                           RCS5 193
C     THIS SECTION USED TO COMPUTE CONE RETURN NEAR SPECULAR TO           RCS5 194
C                       CONIC SURFACE  (THET NEAR (PI/2)-ALPHA)           RCS5 195
      XX        = CMPLX(0.0,1.0)                                          RCS5 196
      ZZ        = XOMX12 * XX * TAUSQ2 * SIN(APT) /3.0                    RCS5 197
      YY        =-X2PX12 * FSTCOP                                         RCS5 198
      FFVV      =-PISQRT * A2 * PHASE * ( YY + ZZ)                        RCS5 199
      FFHH      = PISQRT * A2 * PHASE * ( YY - ZZ)                        RCS5 200
      GO TO 55                                                            RCS5 201
C                                                                         RCS5 202
  130 FFVV    = -         PISQRT * (TERM1V + TERM2V + TERM3V + TERM4V +   RCS5 203
     1                             TERM5V + TERM6V + TERM7V + TERM8V )    RCS5 204
      FFHH    =           PISQRT * (TERM1H + TERM2H + TERM3H + TERM4H +   RCS5 205
     1                             TERM5H + TERM6H + TERM7H + TERM8H )    RCS5 206
C                                                                       * RCS5 207
   55 CONTINUE                                                            RCS5 208
      FFVV       = FFVV * 0.02540005                                      RCS5 209
      FFHH       = FFHH * 0.02540005                                      RCS5 210
C                                                                         RCS5 211
      EVVR(I) = REAL (FFVV)                                               RCS5 212
      EVVI(I) =-AIMAG(FFVV)                                               RCS5 213
      EHHR(I) = REAL (FFHH)                                               RCS5 214
      EHHI(I) =-AIMAG(FFHH)                                               RCS5 215
C                                                                       * RCS5 216
      IF (KP)  77, 77, 78                                                 RCS5 217
      FHHR(I) = REAL (FFHH)                                              RCS5 214
      FHHI(I) =-AIMAG(FFHH)                                              RCS5 215
C                                                                      * RCS5 216
      IF (KP)  77, 77, 78                                                RCS5 217
   78 FREQ(I)  = XI * DF / 1000.0                                        RCS5 218
      SIGMAV(I)= 10.0*ALOG10(FVVR(I)*FVVR(I) + FVVI(I)*FVVI(I))          RCS5 219
      SIGMAH(I)= 10.0*ALOG10(FHHR(I)*FHHR(I) + FHHI(I)*FHHI(I))          RCS5 220
   77 CONTINUE                                                           RCS5 221
C                                                                      * RCS5 222
  200 CONTINUE                                                           RCS5 223
C                                                                      * RCS5 224
C                                                                      * RCS5 225
C     IF (KP)  75, 75, 76                                                RCS5 226
C  76 WRITE (6, 74) ( FREQ(I), SIGMAV(I), SIGMAH(I), I= NMIN, NMAX )     RCS5 227
C  74 FORMAT (1H0, 3E15.5 )                                              RCS5 228
   75 CONTINUE                                                           RCS5 229

C                                                                      * RCS5 230
C                                                                      * RCS5 231
      RETURN                                                             RCS5 232
      END                                                                RCS5 233
```

3

```
      FUNCTION FIRS(XN)                                                 RCS5 234
      PI     = 3.14159265358979                                         RCS5 235
      A      = (SIN(PI / XN)) / XN                                      RCS5 236
      B      = 1.0 / (COS(PI / XN) - 1.0)                               RCS5 237
      FIRS   = A * B                                                    RCS5 238
      RETURN                                                            RCS5 239
      END                                                               RCS5 240
      FUNCTION SECC(PHI, XN)                                            RCS5 241
      PI     = 3.14159265358979                                         RCS5 242
      A      = (SIN(PI / XN)) / XN                                      RCS5 243
      B      = COS(PI / XN) - COS((2.0 * PHI)/ XN)                      RCS5 244
      C      = 1.0 / B                                                  RCS5 245
      SECC   = A * C                                                    RCS5 246
      RETURN                                                            RCS5 247
      END                                                               RCS5 248
      FUNCTION Q(Z)                                                     RCS5 249
C     Q(Z) = 0.5*(1 + ERF(Z))                                           RCS5 250
C   * ERF(Z) IS EVALUATED USING A RATIONAL POLYNOMIAL APPROXIMATION     RCS5 251
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,            RCS5 252
C   *           SECTION 7.1.26)                                         RCS5 253
C                                                                       RCS5 254
      IF ( Z.GT. 2.) GO TO 10                                           RCS5 255
      IF ( Z.LT.-2.) GO TO 20                                           RCS5 256
      AZ = ABS(Z)                                                       RCS5 257
      P = 1.0/(1.0 + .47047*AZ)                                         RCS5 258
      IF ( Z.GT. 2.) GO TO 10                                          RCS5 255
      IF ( Z.LT.-2.) GO TO 20                                          RCS5 256
      AZ = ABS(Z)                                                      RCS5 257
      P = 1.0/(1.0 + .47047*AZ)                                        RCS5 258
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)   RCS5 259
      IF (Z) 2,4,6                                                     RCS5 260
    2 Q = (1.0 - Y)/2.                                                 RCS5 261
      RETURN                                                           RCS5 262
    4 Q = .5                                                           RCS5 263
      RETURN                                                           RCS5 264
    6 Q = (1.0 + Y)/2.                                                 RCS5 265
      RETURN                                                           RCS5 266
   10 Q = 1.                                                           RCS5 267
      RETURN                                                           RCS5 268
   20 Q = 0.                                                           RCS5 269
      RETURN                                                           RCS5 270
      END                                                              RCS5 271
      SUBROUTINE BESL ( X, B0, B1, B2 )                                RCS5 272
C                                                                      RCS5 273
C    * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS  RCS5 274
C    * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS                RCS5 275
C    * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )RCS5 276
C                                                                      RCS5 277
      S = 1.0                                                          RCS5 278
      IF (X .LT. 0.0) S = -1.0                                         RCS5 279
      X = ABS (X)                                                      RCS5 280
C                                                                      RCS5 281
      IF ( X .GT. 1.E-6 ) GO TO 5                                      RCS5 282
      6 Q = (1.0 + Y)/2.                                               RCS5 265
        RETURN                                                         RCS5 266
     10 Q = 1.                                                         RCS5 267
        RETURN                                                         RCS5 268
     20 Q = 0.                                                         RCS5 269
        RETURN                                                         RCS5 270
        END                                                            RCS5 271
        SUBROUTINE BESL ( X, B0, B1, B2 )                              RCS5 272
C                                                                      RCS5 273
C   * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS   RCS5 274
C   * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS                 RCS5 275
C   * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )RCS5 276
C                                                                      RCS5 277
        S = 1.0                                                        RCS5 278
        IF (X .LT. 0.0) S = -1.0                                       RCS5 279
        X = ABS (X)                                                    RCS5 280
C                                                                      RCS5 281
        IF ( X .GT. 1.E-6 ) GO TO 5                                    RCS5 282
        B0 = 1.0                                                       RCS5 283
        B1 = 0.0                                                       RCS5 284
        B2 = 0.0                                                       RCS5 285
        X = X * S                                                      RCS5 286
        RETURN                                                         RCS5 287
```

L-49d

L-49e

```
   CARD NO          ****                               CONTENTS                                        ****

     580            C                                                                          RCS5 288
     581                5 CONTINUE                                                             RCS5 289
     582            C                                                                          RCS5 290
     583                1 IF ( X .GE. 3.) GO TO 9                                              RCS5 291
     584                  X1 = X/3.                                                            RCS5 292
     585                  X1 = X1*X1                                                           RCS5 293
     586                  B = 1.+ X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479RCS5 294
     587                 1 +       X1*(-.0039444+ X1*2.1E-4 )))) )                             RCS5 295
     588                  GO TO 10                                                             RCS5 296
     589            C                                                                          RCS5 297
     590                9 X2 = 3./X                                                            RCS5 298
     591                  FO =  .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*  RCS5 299
     592                 1     (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )                RCS5 300
     593                  TO = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573  RCS5 301
     594                 1    +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))            RCS5 302
     595                  B = FO*COS(TO)/SQRT(X)                                               RCS5 303
     596            C                                                                          RCS5 304
     597               10 BO = B                                                               RCS5 305
     598            C                                                                          RCS5 306
     599                2 IF ( X .GE. 3. ) GO TO 19                                            RCS5 307
     600                  X1 = X/3.                                                            RCS5 308
     601                  X1 = X1*X1                                                           RCS5 309
     1     +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))             RCS5 302
      B = F0*COS(T0)/SQRT(X)                                                RCS5 303
C                                                                           RCS5 304
   10 B0 = B                                                                RCS5 305
C                                                                           RCS5 306
    2 IF ( X .GE. 3. ) GO TO 19                                             RCS5 307
      X1 = X/3.                                                             RCS5 308
      X1 = X1*X1                                                            RCS5 309
      B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*        RCS5 310
     1                (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )       RCS5 311
      GO TO 20                                                              RCS5 312
C                                                                           RCS5 313
   19 X2 = 3./X                                                             RCS5 314
      F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*       RCS5 315
     1     (-.00249511 +X2*(.00113653 -.00020033*X2 )))))                   RCS5 316
      T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879      RCS5 317
     1    +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))                RCS5 318
      B = F1*COS(T1)/SQRT(X)                                                RCS5 319
C                                                                           RCS5 320
   20 B1 = B * S                                                            RCS5 321
      X = X * S                                                             RCS5 322
      B2= (2./X)*B1 - B0                                                    RCS5 323
   50 RETURN                                                                RCS5 324
      END                                                                   RCS5 325
```

2

```
      COMPLEX FUNCTION F(TAU)                                          RCS5 326
C                                                                      RCS5 327
C     COMPUTES FTAU WHERE FTAU =(EXP(-J*TAU**2)/2*TAU)*SQRT(PI/2.)*    RCS5 328
C                                 (C2(TAU**2) + J*S2(TAU**2))          RCS5 329
C                                                                      RCS5 330
      COMPLEX J, FP                                                    RCS5 331
      PI = 3.14159265358979                                            RCS5 332
      PIO2 = PI/2.                                                     RCS5 333
      C1 = SQRT(PI/2.)                                                 RCS5 334
      C2 = 1./C1                                                       RCS5 335
      ATAUS = ABS(TAU)                                                 RCS5 336
      IF (ATAUS .LE. 0.5  )GO TO 20                                    RCS5 337
C                                                                      RCS5 338
C     FOR TAUS .GT. 0.5, FUNCTION COMPUTED USING POLYNOMIAL            RCS5 339
C    * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,          RCS5 340
C    *                SECTIONS  7.3.9,7.3.10,7.3.32,7.3.33)            RCS5 341
      TAUS = SQRT(ATAUS)                                               RCS5 342
      X = C2*TAUS                                                      RCS5 343
      XS = X*X                                                         RCS5 344
C                                                                      RCS5 345
```

```
10/26/78        INPUT LISTING                       AUTOFLOW CHART SET - FWO/SCL  RADSIM

  CARD NO        ****                        CONTENTS                                    ****

    638               FX = (1.0+0.926*X)/(2.0+1.792*X+3.104*XS)                      RCS5 346
    639               GX = 1.0/(2.0+4.142*X+3.492*XS+6.67*X*XS)                      RCS5 347
    640          C                                                                   RCS5 348
    641               CC1XS = COS(ATAUS)                                             RCS5 349
    642               SC1XS = SIN(ATAUS)                                             RCS5 350
    643          C                                                                   RCS5 351
    644               CX = 0.5 + FX*SC1XS - GX*CC1XS                                 RCS5 352
    645               SX = 0.5 - FX*CC1XS - GX*SC1XS                                 RCS5 353
    646          C                                                                   RCS5 354
    647               IF (TAU .LT. 0.0) GO TO 10                                     RCS5 355
    648               B = CMPLX(CX,SX)                                               RCS5 356
    649               FP = CMPLX(  COS(ATAUS), -SIN(ATAUS) )                         RCS5 357
    650               F = (C1*B*FP)/TAUS                                             RCS5 358
    651               RETURN                                                         RCS5 359
    652          C                                                                   RCS5 360
    653            10 CONTINUE                                                       RCS5 361
    654               B = CMPLX(SX,CX)                                               RCS5 362
    655               A = ATAUS-PI02                                                 RCS5 363
    656               FP = CMPLX( COS(A),SIN(A) )                                    RCS5 364
    657               F = (B*FP*C1)/TAUS                                             RCS5 365
    658               RETURN
```

```
      SX = 0.5 - FX*CC1XS - GX*SC1XS                                     RCS5 353
C                                                                        RCS5 354
      IF (TAU .LT. 0.0) GO TO 10                                         RCS5 355
      B = CMPLX(CX,SX)                                                   RCS5 356
      FP = CMPLX(  COS(ATAUS), -SIN(ATAUS) )                             RCS5 357
      F = (C1*B*FP)/TAUS                                                 RCS5 358
      RETURN                                                             RCS5 359
C                                                                        RCS5 360
   10 CONTINUE                                                           RCS5 361
      B = CMPLX(SX,CX)                                                   RCS5 362
      A = ATAUS-PIO2                                                     RCS5 363
      FP = CMPLX( COS(A),SIN(A) )                                        RCS5 364
      F = (B*FP*C1)/TAUS                                                 RCS5 365
      RETURN                                                             RCS5 366
C                                                                        RCS5 367
   20 CONTINUE                                                           RCS5 368
C     FOR TAUS .LE. 0.5, FUNCTION IS EXPANDED IN SERIES AND FIRST FEW    RCS5 369
C                               TERMS INTEGRATED TERM BY TERM TO OBTAIN RESULT RCS5 370
      FP = CMPLX(COS(TAU),-SIN(TAU))                                     RCS5 371
      TS = TAU*TAU                                                       RCS5 372
      FR = 1 - TS*( .1 - .0046296296*TS)                                 RCS5 373
      FI = TAU *( .333333333 - TS*(.0238095238 - 7.5757575757E-4*TS))    RCS5 374
      B = CMPLX(FR.FI)                                                   RCS5 375
      F = FP*B                                                           RCS5 376
      RETURN                                                             RCS5 377
      END                                                                RCS5 378
```

3

L-49f

L.4 THIN WIRE

The far-field scattering from a thin wire has been programmed using the formulation of Ufimtsev (Ref. 4 and 5). The expression includes the higher order scattering terms which arise from waves which are launched from one end of the wire, traverse the wire length, and are diffracted into space and reflected along the wire upon reaching the opposite end. These scattering mechanisms and the geometry of the problem are shown in Figure L.4-1. The ray components of the scattered field are described in References 4 and 5. The expression of the scattered field, which includes only the horizontal polarization response, is the following:

$$e(\theta) = \frac{2\sqrt{\pi}}{k} \bullet \frac{2iS(\theta)}{\sin\theta \sin 2\theta \left[\ln\left(\frac{2i}{\gamma ka \sin\theta}\right)\right]^2}$$

where $\gamma = 1.781$

$k = 2\pi/\lambda =$ wave number

$$\begin{aligned} S(\theta) = & -\left\{\sin^4(\theta/2)\ln\left(\frac{i}{\gamma ka \sin^2\theta/2}\right)\right\} \\ & + \left\{e^{i2kL\cos\theta}\cos^4(\theta/2)\ln\left(\frac{i}{\gamma ka \cos^2\theta/2}\right)\right. \\ & - e^{ikL(1+\cos\theta)}\cos^4(\theta/2)\ln\left(\frac{i}{\gamma ka \cos\theta/2}\right)2\Psi_+ \\ & + e^{ikL(1+\cos\theta)}\sin^4(\theta/2)\ln\left(\frac{i}{\gamma ka \sin\theta/2}\right)2\Psi_- \\ & \left. + e^{i2kL}\cos\theta\ln(i/\gamma ka)(\Psi_+)^2\right\} \\ & + \left\{\left[\sum_{n=0}^{\infty} e^{i2nkL\Psi 2n}\right]\left[(\Psi)^2(\Psi_+)^2 e^{i4kL}\right.\right. \\ & + (\Psi_-)^2 e^{ikL2(1+\cos\theta)} \\ & \left.\left. - 2\Psi_+\Psi_-\Psi e^{ikL(3+\cos\theta)}\right]\cos\theta\ln\left(\frac{i}{\gamma ka}\right)\right]\right\} \end{aligned}$$

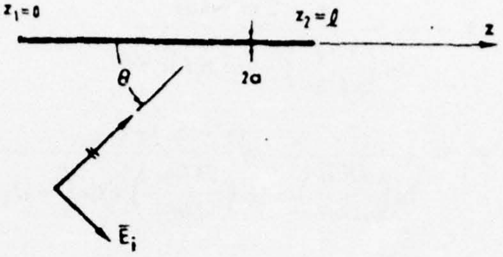
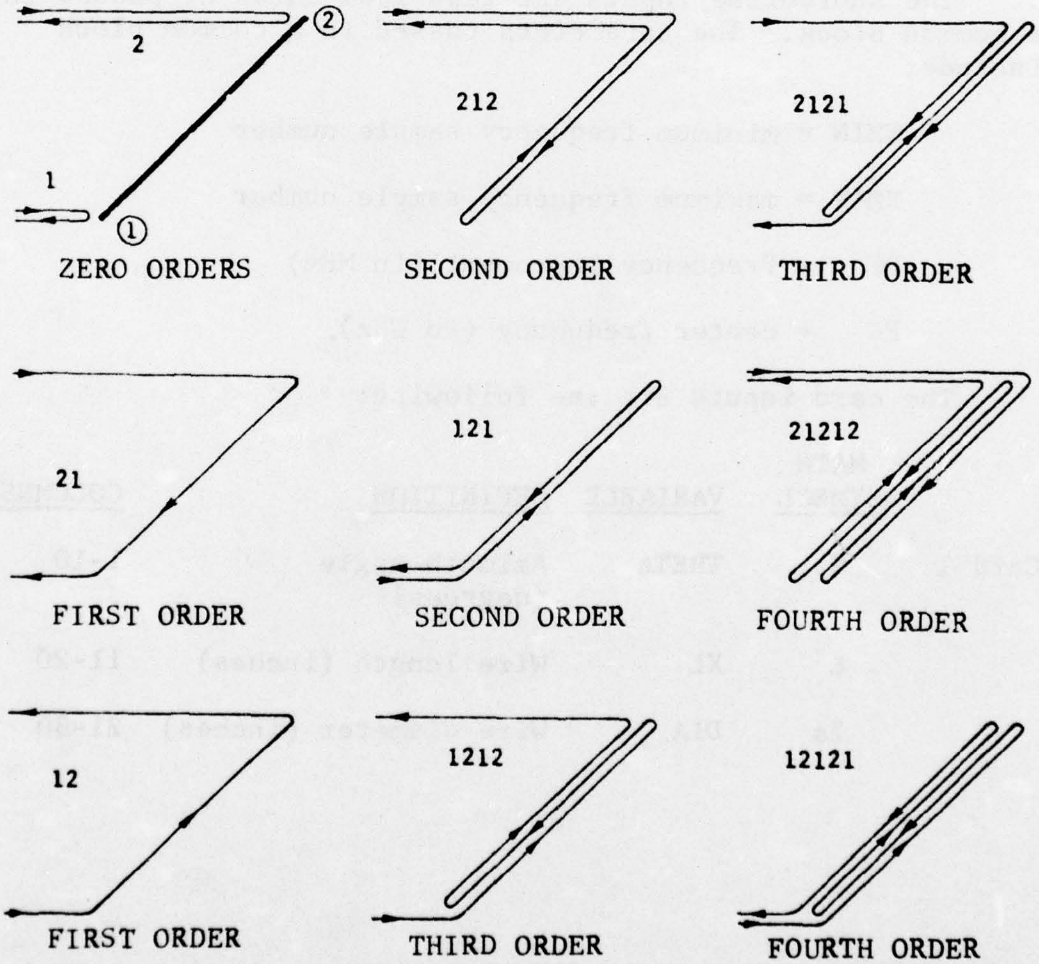


Fig. L.4-1a GEOMETRY OF A THIN WIRE



ZERO ORDERS

SECOND ORDER

THIRD ORDER

FIRST ORDER

SECOND ORDER

FOURTH ORDER

FIRST ORDER

THIRD ORDER

FOURTH ORDER

Fig. L.4-1b RAY COMPONENTS OF THE SCATTERED RETURNS

and

$$\Psi = \frac{i\pi - 2\ln(\gamma ka)}{\ln\left(\frac{i2kL}{\gamma k^2a^2}\right) - E(2kL)e^{-i2kL}}$$

$$\Psi_\pm = \frac{i\pi - \ln(\gamma^2 q_\pm)}{\ln\left(\frac{i2kL}{\gamma k^2a^2}\right) - E\left(\frac{2kLq_\pm}{k^2a^2}\right)\exp\left(-i2q_\pm\frac{kL}{k^2a^2}\right)}$$

$$q_\pm = \frac{(ka)^2}{2}(1 \mp \cos\theta)$$

$$E(x) = \int_\infty^x \frac{e^{it}}{t}\,dt.$$

L.4.1 Inputs

The subroutine inputs are read from cards or passed in a common block. The parameters passed in a common block include:

NMIN = minimum frequency sample number

NMAX = maximum frequency sample number

DF = frequency increment (in MHz)

FC = center frequency (in GHz).

The card inputs are the following:

	MATH SYMBOL	VARIABLE	DEFINITION	COLUMNS
Card 1	θ	THETA	Azimuth angle (degrees)	1-10
	L	XL	Wire length (inches)	11-20
	2a	DIA	Wire diameter (inches)	21-30

L.4.2 Outputs

The data base output consists of two linear arrays, ETTR, ETTI, which contain the real and imaginary parts of the horizontally polarized backscattered fields (in meters) at frequency increments spaced DF MHz from NMIN*DF to NMAX*DF.

If selected by setting KP to 1, the wire radar cross section (in dBsm) versus frequency will be printed out. In addition the real and imaginary parts of the exponential integral, computed in subroutine EXPI, will be printed out if KP ≠ 0.

L.4.3 Restrictions

L.4.3.1 Physical Dimensions

The wire length should be large with respect to the largest wavelength of the illuminating field. However, the wire radius should be much smaller than the smallest wavelength of the illuminating field.

L.4.3.2 Output

The output arrays are passed in the argument list and a value is computed and stored only in location NMIN to NMAX.

L.4.3.3 Azimuth

The azimuth angle is restricted to the region from 0.5 to 89.5 degrees.

L.4.4 Definition of Selected Terms Used in Subroutine

$$\mathrm{ST1} = -\left\{\sin^4(\theta/2)\ln\left(\frac{i}{\gamma ka\sin^2\theta/2}\right)\right\}$$

$$\mathrm{SCD1} = \left\{e^{i2kL\cos\theta}\cos^4(\theta/2)\ln\left(\frac{i}{\gamma ka\cos^2\theta/2}\right)\right.$$

$$\mathrm{SCD2} = -e^{ikL(1+\cos\theta)}\cos^4(\theta/2)\ln\left(\frac{i}{\gamma ka\cos\theta/2}\right)2\Psi_{+}$$

SCD3 = $+ e^{ikL(1+\cos\theta)} \sin^4(\theta/2) \ln\left(\frac{i}{\gamma ka \sin\theta/2}\right) 2\Psi$

SCD4 = $+ e^{i2kL} \cos\theta \ln(i/\gamma ka)(\Psi_+)^2$

$$D = \frac{1}{1 - \psi^2(L)\, e^{i2kL}} = \left[\sum_{n=0}^{\infty} e^{i2nkL\Psi 2n}\right]$$

SSS =

$$+\left\{\left[\sum_{n=0}^{\infty} e^{i2nkL\Psi 2n}\right]\left[(\Psi)^2(\Psi_+)^2 e^{i4kL}\right.\right.$$
$$+ (\Psi_-)^2 e^{ikL2(1+\cos\theta)}$$
$$\left.\left.- 2\,\Psi_+\Psi_-\Psi\, e^{ikL(3+\cos\theta)}\right]\cos\theta \ln\left(\frac{i}{\gamma ka}\right)\right\}$$

PSIA = $$\Psi = \frac{i\pi - 2\ln(\gamma ka)}{\ln\left(\frac{i2kL}{\gamma k^2a^2}\right) - E(2kL)e^{-i2kL}}$$

PSI $\begin{Bmatrix} P \\ M \end{Bmatrix}$ = $$\Psi_\pm = \frac{i\pi - \ln(\gamma^2 q_\pm)}{\ln\left(\frac{i2kL}{\gamma k^2a^2}\right) - E\left(\frac{2kLq_\pm}{k^2a^2}\right)\exp\left(-i2q_\pm\frac{kL}{k^2a^2}\right)}$$

L.4.5 Subroutines

The subroutine EXPI (ARGZ, EIXR, EIXI, KP) is used to compute the real (EIXR) and imaginary (EIXI) parts of the exponential integral:

$$E(x) = \int_{\infty}^{x} \frac{e^{t}}{t}\,dt$$

where the argument X is passed as ARGZ.

RCSM1   4.4, 4.5, 4.6

```
6     16.87


$      IDENT    BECAGD01,HANCOCK,01707310038 0,RCSM1
$      OPTION   FORTRAN
$      FORTY    LSTIN,XREF,MAP,DECK
$      LIMITS   05,39K,0,5K
      SUBROUTINE  TARGET  (ETTR, ETTI,XR,XI)
C
C     **  THIN WIRE CW RESPONSE    * 0.5 TO 89.5 DEGREES ASPECT ANGLE**
C     *   SOLUTION BY UF1MTSEV, SIMPLIFIED BY HONG FOR BACKSCATTER     *
C   * REFERENCE - SHORT PULSE SCATTERING BY A LONG WIRE,  BY S HONG
C   *               IEEE ON AP, VOL AP-16,NO.3, MAY 1968,PP.338-342
C
C
      COMMON MOVER,M,NMIN, NMAX, DF, FC, PW, T0
C     NMIN = MINIMUM FREQUENCY SAMPLE
C     NMAX = MAXIMUM FREQUENCY SAMPLE
C     DF   = FREQUENCY INCREMENT IN MHZ
C     FC   = CARRIER FREQUENCY IN GHZ
C
      COMPLEX     SS(100),
     A          ACLC,ALN1,  CS,     ACL2C, ST1, EXA,EXQP,EXQM,EXAP,PSIA,
     B         EXQPP,     EXQMP,PSIP,PSIM, AC1, PC1, AC2,SCD1, BC1, QC1,
     C          SCD2, DC1,SCD3, RC1, EC1,SCD4, SCD, FP8, FP9,FP10,
     D           SF1, SF2, SF3,  SF,  SP, SPC, SSS, BSC,EBSC
     E                , PIJ,    PIJO2
     F        ,CC6RL,   D   ,SB4
     G                ,EXQA,EXQM
     1  , RC9, QC9
      REAL  FRQ(512), ETTR(512), ETTI(512), SIGMA(512),XR(512),XI(512)
C
      READ(5,1000) THETA, XL, DIA, KP
C     KP = 1, PRINT OUT RCS VERSUS FREQUENCY
 1000 FORMAT (3F10.3, I2)
      WRITE(6,1100) THETA, XL, DIA
 1100 FORMAT ('0 ASPECT ANGLE = ',F7.2,'    WIRE LENGTH = ', F8.3, //,
     A  '   WIRE DIAMETER  = ', F7.4 , // , ' ( LENGTHS ARE IN INCHES )'
     B)
      IF (THETA .LT. 0.5  .OR. THETA .GT. 89.5 ) GO TO 900
      RAD = DIA/2.0
      PI = 3.1415926
      PIJ = CMPLX(0.0,PI)
      PIO2= PI/2.0
      PIJO2 = CMPLX(0.0,PIO2)
      GAM = 1.781072
      TH  = THETA * (180.0 / PI)
      CT = COS(TH)
      CTO2 = COS(TH/2.0)
      CTO2S= CTO2 * CTO2
      CTO2F= CTO2S * CTO2S
      ST = SIN (TH)
      STT = SIN(TH*2.0)
      STO2 = SIN(TH/2.0)
      STO2S = STO2 * STO2
      STO2F = STO2S * STO2S
      OPCT = 1.0+ CT
      OMCT = 1.0- CT
      AKTH = 2.0/(ST*STT)
C
      DO 800  IFW = NMIN,NMAX
      XI= IFW - 1
      FREQ    = XI * DF / 1000.0
      FRQ( IFW) = FREQ
      XK =  ( .53234454 * FREQ )
      XKL =  XK* XL
      XKA =  XK* RAD
      GKA =  GAM* XKA
      C1 = 2.0/( GKA * ST)
      ACL = ALOG ( C1 )
      ACLC =  CMPLX ( ACL, PIO2 )
      CS =((0.0, 1.0)/(ACLC*ACLC))*(-AKTH)
C
C     ** DETERMINATION OF INTEGRAL AND OTHER TERMS USED IN MULTIPLE **
C        ORDER SCATTERING
C
      C5L = ALOG( GKA)
      C6R = (2.0/XKA)*(XKL/GKA)
      C6RL= ALOG( C6R)
      CC6RL = CMPLX(C6RL,PIO2)
      EAA = XKL+2.0
      CALL EXPI ( EAA, EXRA, EXIA, 0 )
      EXQA = CMPLX( EXRA, EXIA)
      E1C= COS( EAA)
      E1S= SIN( EAA)
      RC1 = CMPLX (E1C, E1S)
      RC9      = CMPLX( E1C, -E1S )
      PSIA     = (PIJ - 2.0*C5L) / (CC6RL - EXQA*RC9)
C
      EQPA = XKL*OMCT
      CALL EXPI (EQPA, EXR, EXI, 0 )
      EXQP= CMPLX( EXR, EXI )
      E2C =COS( EQPA)
      E2S =SIN( EQPA)
      EXQPP = CMPLX( E2C,-E2S)
      C7P = GKA+GKA*OMCT/2.0
      C7PL = ALOG(C7P)
      PSIP = (PIJ - C7PL)/(CC6RL - EXQP*EXQPP)
C
      EQMA = XKL*OPCT
      CALL EXPI ( EQMA, EXRM, EXIM, 0 )
      EXQM = CMPLX( EXRM, EXIM)
      Q2       = COS(EQMA)
      Q3       = SIN(EQMA)
      QC1 = CMPLX( Q2, Q3)
      QC9      = CMPLX( Q2, -Q3 )
      C7M = GKA*GKA*OPCT/2.0
      C7ML = ALOG(C7M)
      PSIM     = (PIJ - C7ML) / (CC6RL - EXQM*QC9)
C
C     * UTIL12ATION OF FACTORS IN SCATTERED FIELD EXPRESSION *
C
C     **  FIRST ORDER SCATTERING   **
C
      C2 = 1.0 / (GKA * STO2S)
      ACL2 = ALOG (C2)
      ACL2C = CMPLX ( ACL2, PIO2 )
      ST1 = -STO2F * ACL2C
C
C     *   RETURN FROM TRAILING EDGE OF WIRE   *
C
      A1    = 1.0/(GKA* CTO2S)
      A2    = ALOG(A1)
      AC1   = CMPLX( A2,PIO2)
      P1 =  2.0*XKL*CT
      P2 = COS(P1)
      P3 = SIN(P1)
      PC1= CMPLX( P2, P3)
      AC2 = CMPLX( CTO2F,0.0)
      SCD1 = AC1*PC1*AC2
C
```

```
      B1 = 1.0/(GKA*CTO2)
      B2 = 2.0*ALOG(B1)
      BC1 = CMPLX( B2,PI)
      SCD2 =(-(QC1*AC2*BC1*PSIP))
C
      D1 = 1.0/(GKA*STO2)
      D2 = 2.0*ALOG(D1)
      DC1= CMPLX(D2,PI)
      SB4      = CMPLX( STO2F, 0.0)
      SCD3 = SB4  *(QC1*DC1*PSIM)
C
      E1 = 1.0/GKA
      E2 = ALOG( E1)
      EC1= CMPLX( E2,PIO2)
      SCD4 = CT*(RC1*EC1*PSIP*PSIP)
C
      SCD = SCD1 + SCD2 + SCD3 + SCD4
C
      FP1 = COS(4.0*XKL)
      FP2 = SIN(4.0*XKL)
      FP3 = COS(2.0*XKL*OPCT)
      FP4 = SIN(2.0*XKL*OPCT)
      FP5 = XKL*(3.0+CT)
      FP6 = COS(FP5)
      FP7 = SIN(FP5)
      FP8  = CMPLX( FP1,FP2)
      FP9  = CMPLX( FP3,FP4)
      FP10 = CMPLX( FP6,FP7)
C
      SF1 = PSIA*PSIA*PSIP*PSIP*FP8
      SF2 = PSIM*PSIM*FP9
      SF3 = -2.0*PSIP*PSIM*PSIA*FP10
      SF = (SF1+SF2+SF3)*CT*EC1
C
      D =1.0-(PSIA*PSIA)* RC1
      AD =CABS(D)
      IF (AD .LE. 1.0E-6 ) GO TO 213
      SSS =  SF/D
      GO TO 221
  213 WRITE (6, 2005)
 2005 FORMAT ('0 DENOMINATOR IS ZERO . PSIA IS TOO LARGE')
      GO TO 999
C
C     * BACKSCATTERED FIELD   *
C
  221 BSC = (ST1 + SCD + SSS)* CS
```

```
      CK1 = (4.0* PI)/(XK*XK)
      CK2 = SQRT(CK1)*0.02540
C
C     **  EBSC = SQRT(SIGMA) WITH PHASE REFERENCED TO FRONT   **
C      *                EDGE   OF   WIRE                       *
      EBSC =(BSC* CK2)
      ETTR(IFW)= REAL(EBSC)
       XR(IFW) = ETTR(IFW)
      ETTI(IFW)=-AIMAG(EBSC)
       XI(IFW) = ETTI(IFW)
  800 CONTINUE
C
      IF (KP .NE. 1)  GO TO 900
      DO 777  L = NMIN, NMAX
      SIGMA(L)  = 10.0 * ALOG10(ETTR(L)*ETTR(L) + ETTI(L)*ETTI(L))
  777 CONTINUE
C     WRITE(6,3000) (FRQ(J), SIGMA(J), J = NMIN, NMAX)
C3000 FORMAT ( '1FREQUENCY RESPONSE OF A THIN WIRE '///,   ' FREQUENCY
C    1   CROSS SECTION    ', //, (E12.4,  E15.4) )
  900 CONTINUE
  999 CONTINUE
      RETURN
      END
      SUBROUTINE EXPI ( ARGZ, EIXR, EIXI, KP)
C
C   * THIS SUBROUTINE COMPUTES THE REAL AND IMAGINARY PARTS OF THE
C   *       EXPONENTIAL INTEGRAL E(X) WHERE
C   *       E(X) =  INTEGRAL FROM X TO INFINITY OF EXP(IT)/T*DT
C   * REFERENCE - FUNCTIONS OF MATHEMATICAL PHYSICS BY MAGNUS AND
C   *                OBERHETTINGER, PP.97-98
C     KP .NE. 0, PRINT OUT REAL AND IMAGINARY PARTS OF EXPONENTIAL
C               INTEGRAL
      COMPLEX  AIX,       ANTG( 100),          EIX,       FA,         F
      REAL  AINTR(100), AINTI(100)
C
      DEL = 0.000001
      GAMMA = 0.57721566
      IF ( ARGZ .LE. 1.0E-6) GO TO 50
      ABZ = ABS(ARGZ)
      DO 5  I=1,100
      AINTR(I)= 0.0
      AINTI(I)= 0.0
      ANTG(I) = CMPLX(0.0,0.0)
    5 CONTINUE
      IF (  ABZ .GE. 15) GO TO 9
C
C     THIS SERIES USED FOR  0.LT.ABS(X).LT 15
C     **  SERIES EXPANSION INVOLVING CI(X) AND SI(X)   **
C
C     E(2) =  GAM    + LN(Z) +SUM(((-1)**N)*Z**(2*N))/((2*N)*FACT(2*N))
C           +J*(     *SUM(((-1)**N)*Z**(2*N+1))/((2*N+1)*FACT(2*N+1))-
C           PIO2 )
C
      INC =1
      FAC = -(ARGZ**2)/(2.0)
      AINTR(1) =  FAC * 0.5
      AINTI(1) = FAC*(ARGZ/9.0) + ARGZ
C
   10 INC = INC + 1
      X2 =(2* INC)
      X2M = X2 -1.0
      X2P1S = (2*INC + 1 )**2
      FAC = -(FAC*ARGZ*ARGZ)/(X2M*X2)
      AINTR(INC) = FAC/X2 + AINTR(INC-1)
      AINTI(INC) = FAC*(ARGZ/X2P1S) +AINTI(INC-1)
CC
      ABR   = ABS( AINTR(INC))
      ABRM1= ABS( AINTR(INC-1))
      ABI   = ABS( AINTI(INC)   )
      ABIM1= ABS( AINTI(INC-1))
      IF ( ABS( ABR - ABRM1) .GE. DEL) GO TO 20
      IF ( ABS( ABI - ABIM1) .LE. DEL) GO TO 40
   20 CONTINUE
C
      IF ( INC .LT. 100 ) GO TO 10
      WRITE (6, 1000)
 1000 FORMAT( '0SERIES DID NOT CONVERGE ' )
      NI = INC
      GO TO 60
   40 NI= INC
   60 CONTINUE
      EIXR = AINTR(NI) + ALOG(ARGZ) + GAMMA
      EIXI = AINTI(NI) - 1.5707963
      GO TO 75
CC
C     **  ASYMPTOTIC SERIES EXPANSION FOR INT FROM INF TO X OF   **
C         (EXP(-JT)/T)*DT
C
    9 CONTINUE
      IF (  ABZ .GE. 150) GO TO 99
C     SERIES EXPANSION USED WHEN 15.LE.ABS(X).LT.150
C    * E(X) = EXP(IX)*(1/IX + 1/(IX)**2 + 2FACT/(IX)**3+...  )
C
      AIX =  CMPLX(0.0,ARGZ )
      FA = 1.0/AIX
      ANTG(1) = FA
      F   = FA*FA
      ANTG(2) = F       + ANTG(1)
      INC = 2
C
  110 INC = INC + 1
      XF = INC-1
      F = F*XF*FA
      ANTG(INC) = F + ANTG(INC-1)
      AB =  CABS (ANTG(INC))
      ABM1 =CABS (ANTG(INC-1))
      IF ( ABS(AB - ABM1) .LT. DEL) GO TO150
      IF ( INC .LE. 29) GO TO110
C
      WRITE ( 6, 1001 )
 1001 FORMAT ( ' SERIES DID NOT CONVERGE ')
      NN =  15
      GO TO 70
  150 NN = INC
   70 CONTINUE
C
      EIX = ANTG( NN )*CEXP(AIX)
      EIXR = REAL(EIX)
      EIXI = AIMAG(EIX)
      GO TO 75

      FUNCTION SET TO ZERO IF ARGUMENT ABS(X).GE.150
   99 EIXR = 0.0
      EIXI = 0.0
   75 CONTINUE

      IF ( KP .EQ. 0) GO TO 90
      WRITE (6,2000) EIXR, EIXI
 2000 FORMAT ( '0REAL PART OF E(X)= ',E15.5,//,' IMAGINARY PART OF E(X)
     A= ', E15.5 )
      GO TO 90
   50 CONTINUE
      WRITE (6, 3000)
 3000 FORMAT ('0ARGZ DID NOT TRANSFER INTO SUBROUTINE OR IS ZERO' )
   90 CONTINUE
      RETURN
      END
```

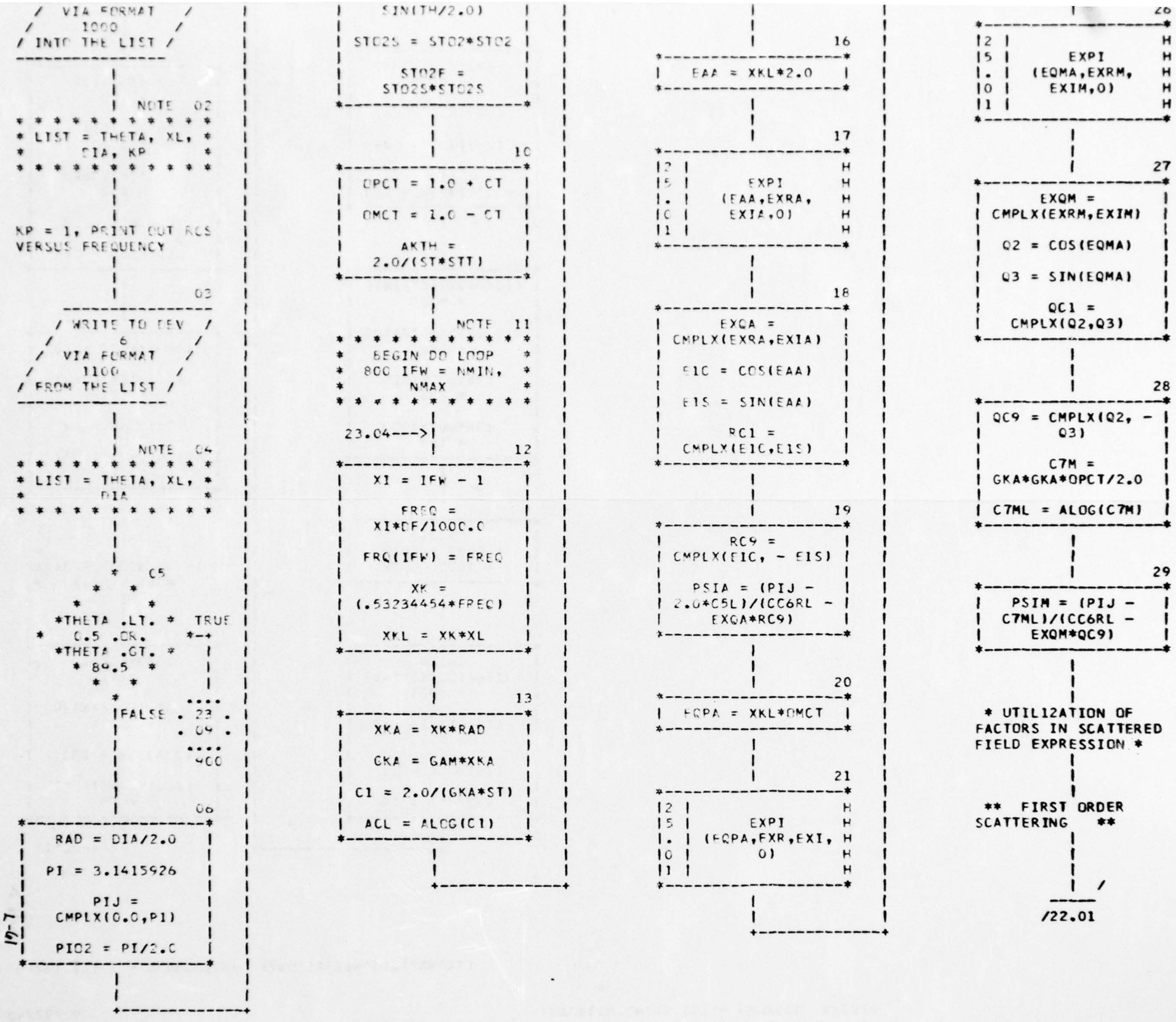
CHART TITLF - SUBROUTINE TARGET(ETTR,ETTI,XR,XI)

```
 -----------
 / TARGET  /
 -----------

      *

 ** THIN WIRE CW
RESPONSE    * C.5 TO
89.5 DEGREES ASPECT
ANGLE**
 *    SOLUTION BY
UFIMTSEV, SIMPLIFIED
BY HONG FOR
BACKSCATTER    *
 * REFERENCE - SHORT
PULSE SCATTERING BY A
LONG WIRE,  BY S HONG
 *              IEEE
ON AP, VOL
AP-16,NO.3, MAY
1968,PP.338-342

NMIN = MINIMUM
FREQUENCY SAMPLE
NMAX = MAXIMUM
FREQUENCY SAMPLE
DF    = FREQUENCY
INCREMENT IN MHZ
FC    = CARRIER
FREQUENCY IN GHZ

        *
        |          01
   ---------------
   / READ FROM DEV /
  /      5        /
 /  VIA FORMAT   /
/     1000      /
/ INTO THE LIST /
---------------
        |
        |
        |  NOTE  02
* * * * * * * * * * * *
* LIST = THETA, XL,  *
*      DIA, KP       *
* * * * * * * * * * * *
        |
        |
        |
KP = 1, PRINT OUT RCS
VERSUS FREQUENCY
        |
```

```
+------------------->*
|                    |          07
|      *-------------------------*
|      |      PIJ02 =            |
|      |  CMPLX(0.0,PI02)        |
|      |                         |
|      |  GAM = 1.781072         |
|      |                         |
|      |        TH =             |
|      |  THETA*(180.0/PI)       |
|      |                         |
|      |    CT = COS(TH)         |
|      *-------------------------*
|                    |
|                    |          08
|      *-------------------------*
|      |      CTO2 =             |
|      |   COS(TH/2.0)           |
|      |                         |
|      | CTO2S = CTO2*CTO2       |
|      |                         |
|      |      CTO2F =            |
|      |   CTO2S*CTO2S           |
|      |                         |
|      |    ST = SIN(TH)         |
|      *-------------------------*
|                    |
|                    |          09
|      *-------------------------*
|      | STT = SIN(TH*2.0)       |
|      |                         |
|      |      STO2 =             |
|      |   SIN(TH/2.0)           |
|      |                         |
|      | STO2S = STO2*STO2       |
|      |                         |
|      |      STO2F =            |
|      |   STO2S*STO2S           |
|      *-------------------------*
|                    |
|                    |          10
|      *-------------------------*
|      |  OPCT = 1.0 + CT        |
|      |                         |
|      |  OMCT = 1.0 - CT        |
|      |                         |
|      |      AKTH =             |
|      |   2.0/(ST*STT)          |
|      *-------------------------*
|                    |
```

```
+------------------->*
|                    |          14
|      *-------------------------*
|      |      ACLC =             |
|      |  CMPLX(ACL,PI02)        |
|      |                         |
|      |       CS =              |
|      |   ((0.0,1.0)            |
|      |   /(ACLC*ACLC))         |
|      |    *(-AKTH)             |
|      *-------------------------*
|                    |
|                    |
|       ** DETERMINATION OF
|      INTEGRAL AND OTHER
|      TERMS USED IN
|      MULTIPLE **
|      ORDER SCATTERING
|                    |
|                    |          15
|      *-------------------------*
|      |  C5L = ALOG(GKA)        |
|      |                         |
|      |       C6R =             |
|      |    (2.0/XKA)            |
|      |    *(XKL/GKA)           |
|      |                         |
|      |  C6RL = ALOG(C6R)       |
|      |                         |
|      |      CC6RL =            |
|      |  CMPLX(C6RL,PI02)       |
|      *-------------------------*
|                    |
|                    |          16
|      *-------------------------*
|      |   EAA = XKL*2.0         |
|      *-------------------------*
|                    |
|                    |          17
|      *-------------------------*
|      |2 |                      H
|      |5 |      EXPI            H
|      |. |   (EAA,EXRA,         H
|      |0 |    EXIA,0)           H
|      |1 |                      H
|      *-------------------------*
|                    |
```

```
+------------------->*
|                    |          23
|      *-------------------------*
|      |      EXQP =             |
|      |  CMPLX(EXR,EXI)         |
|      |                         |
|      |  E2C = COS(EQPA)        |
|      |                         |
|      |  E2S = SIN(EQPA)        |
|      |                         |
|      |      EXQPP =            |
|      | CMPLX(E2C, - E2S)       |
|      *-------------------------*
|                    |
|                    |
|                    |          24
|      *-------------------------*
|      |       C7P =             |
|      | GKA*GKA*OMCT/2.0        |
|      |                         |
|      | C7PL = ALOG(C7P)        |
|      |                         |
|      |   PSIP = (PIJ -         |
|      |  C7PL)/(CC6RL -         |
|      |    EXQP*EXQPP)          |
|      *-------------------------*
|                    |
|                    |          25
|      *-------------------------*
|      | EQMA = XKL*OPCT         |
|      *-------------------------*
|                    |
|                    |          26
|      *-------------------------*
|      |2 |                      H
|      |5 |      EXPI            H
|      |. |  (EQMA,EXRM,         H
|      |0 |    EXIM,0)           H
|      |1 |                      H
|      *-------------------------*
|                    |
|                    |          27
|      *-------------------------*
|      |      EXQM =             |
|      | CMPLX(EXRM,EXIM)        |
|      |                         |
|      |  Q2 = COS(EQMA)         |
|      |                         |
|      |  Q3 = SIN(EQMA)         |
```



```
/ VIA FORMAT /
/ 1000 /
/ INTO THE LIST /

NOTE 02
* LIST = THETA, XL, *
* DIA, KP *

KP = 1, PRINT OUT RCS
VERSUS FREQUENCY

03
/ WRITE TO DEV /
/ 6 /
/ VIA FORMAT /
/ 1100 /
/ FROM THE LIST /

NOTE 04
* LIST = THETA, XL, *
* DIA *

C5
*THETA .LT. *  TRUE
0.5 .OR.
*THETA .GT. *
* 89.5 *
FALSE   . 23 .
        . 09 .
          900

06
RAD = DIA/2.0
PI = 3.1415926
PIJ =
CMPLX(0.0,PI)
PIO2 = PI/2.C
```

```
SIN(TH/2.0)
STO2S = STO2*STO2
STO2F =
STO2S*STO2S

10
OPCT = 1.0 + CT
OMCT = 1.0 - CT
AKTH =
2.0/(ST*STT)

NOTE 11
* BEGIN DO LOOP *
* 800 IFW = NMIN, *
* NMAX *

23.04--->
12
XI = IFW - 1
FREQ =
XI*DF/1000.0
FRQ(IFW) = FREQ
XK =
(.53234454*FREQ)
XKL = XK*XL

13
XKA = XK*RAD
GKA = GAM*XKA
C1 = 2.0/(GKA*ST)
ACL = ALOG(C1)
```

```
16
EAA = XKL*2.0

17
2 | EXPI          H
5 | (EAA,EXRA,    H
. | EXIA,0)       H
0 |               H
1 |               H

18
EXQA =
CMPLX(EXRA,EXIA)
EIC = COS(EAA)
EIS = SIN(EAA)
RC1 =
CMPLX(EIC,EIS)

19
RC9 =
CMPLX(EIC, - EIS)
PSIA = (PIJ -
2.0*C5L)/(CC6RL -
EXQA*RC9)

20
EQPA = XKL*OMCT

21
2 | EXPI              H
5 | (EQPA,EXR,EXI,    H
. | 0)                H
0 |                   H
1 |                   H
```

```
26
2 | EXPI          H
5 | (EQMA,EXRM,   H
. | EXIM,0)       H
0 |               H
1 |               H

27
EXQM =
CMPLX(EXRM,EXIM)
Q2 = COS(EQMA)
Q3 = SIN(EQMA)
QC1 =
CMPLX(Q2,Q3)

28
QC9 = CMPLX(Q2, -
Q3)
C7M =
GKA*GKA*OPCT/2.0
C7ML = ALOG(C7M)

29
PSIM = (PIJ -
C7ML)/(CC6RL -
EXQM*QC9)

* UTILIZATION OF
FACTORS IN SCATTERED
FIELD EXPRESSION *

** FIRST ORDER
SCATTERING **

/22.01
```

L-62

```
04/26/76                                                      AUTOFLOW CHART SET - FWD/SCL  RADSIM

CHART TITLE - SUBROUTINE  TARGET(FTTR,FTTI,XR,XI)

  21.30--->*                  +------------------->*
           |          01      |                    |          07
  *------------------------*  |        *------------------------*
  |         C2 =           |  |        |    E1 = 1.0/GKA        |
  |   1.0/(GKA*STO2S)      |  |        |                        |
  |                        |  |        |    E2 = ALOG(E1)       |
  |   ACL2 = ALOG(C2)      |  |        |                        |
  |                        |  |        |        EC1 =           |
  |        ACL2C =         |  |        |   CMPLX(E2,PIO2)       |
  |   CMPLX(ACL2,PIO2)     |  |        |                        |
  |                        |  |        |        SCD4 =          |
  |        ST1 = -         |  |        |  CT*(RC1*EC1*PSIP*     |
  |     STO2F*ACL2C        |  |        |        PSIP)           |
  *------------------------*  |        *------------------------*
           |                  |                    |
           |                  |                    |
  *    RETURN FROM            |                    |          08
  TRAILING EDGE OF WIRE       |        *------------------------*
  *                           |        |   SCD = SCD1 +         |
           |                  |        |   SCD2 + SCD3 +        |
           |          02      |        |        SCD4            |
  *------------------------*  |        *------------------------*
  |         A1 =           |  |                    |
  |   1.0/(GKA*CTO2S)      |  |                    |          09
  |                        |  |        *------------------------*
  |    A2 = ALOG(A1)       |  |        |        FP1 =           |
  |                        |  |        |    COS(4.0*XKL)        |
  |        AC1 =           |  |        |                        |
  |   CMPLX(A2,PIO2)       |  |        |        FP2 =           |
  |                        |  |        |    SIN(4.0*XKL)        |
  |   P1 = 2.0*XKL*CT      |  |        |                        |
  *------------------------*  |        |        FP3 =           |
           |                  |        |  COS(2.0*XKL*OPCT)     |
           |          03      |        |                        |
  *------------------------*  |        |        FP4 =           |
  |    P2 = COS(P1)        |  |        |  SIN(2.0*XKL*OPCT)     |
  |                        |  |        *------------------------*
  |    P3 = SIN(P1)        |  |                    |
  |                        |  |                    |          10
  |        PC1 =           |  |        *------------------------*
  |    CMPLX(P2,P3)        |  |        |  FP5 = XKL*(3.0 +      |
  |                        |  |        |        CT)             |
  |        AC2 =           |  |        |                        |
  |  CMPLX(CTO2F,0.0)      |  |        |   FP6 = COS(FP5)       |
  |                        |  |        |                        |
  |        SCD1 =          |  |        |   FP7 = SIN(FP5)       |
  |    AC1*PC1*AC2         |  |        |                        |
  *------------------------*  |        |        FP8 =           |
           |                  |        |   CMPLX(FP1,FP2)       |
                                       *------------------------*
                                                   |
```

2

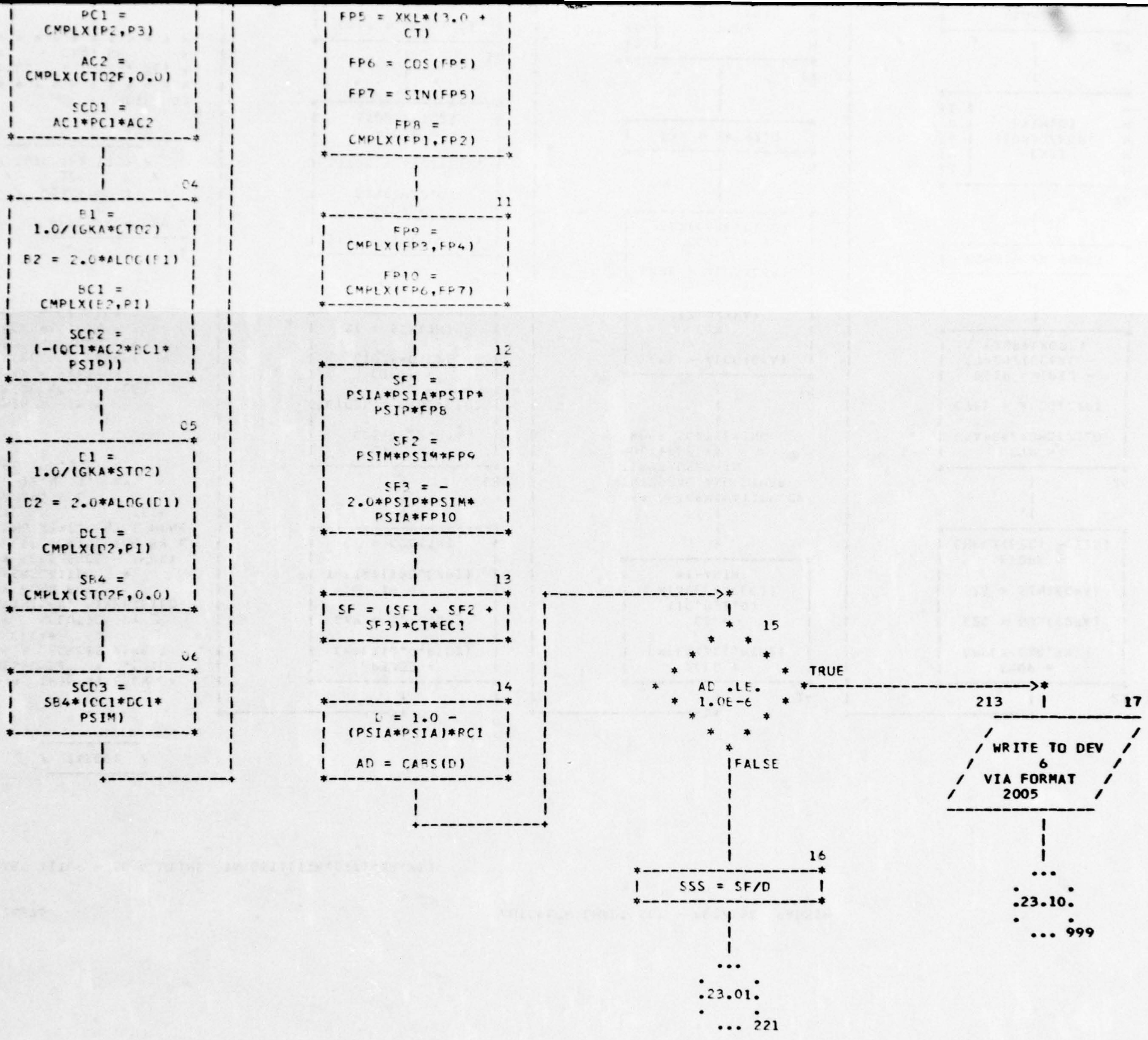
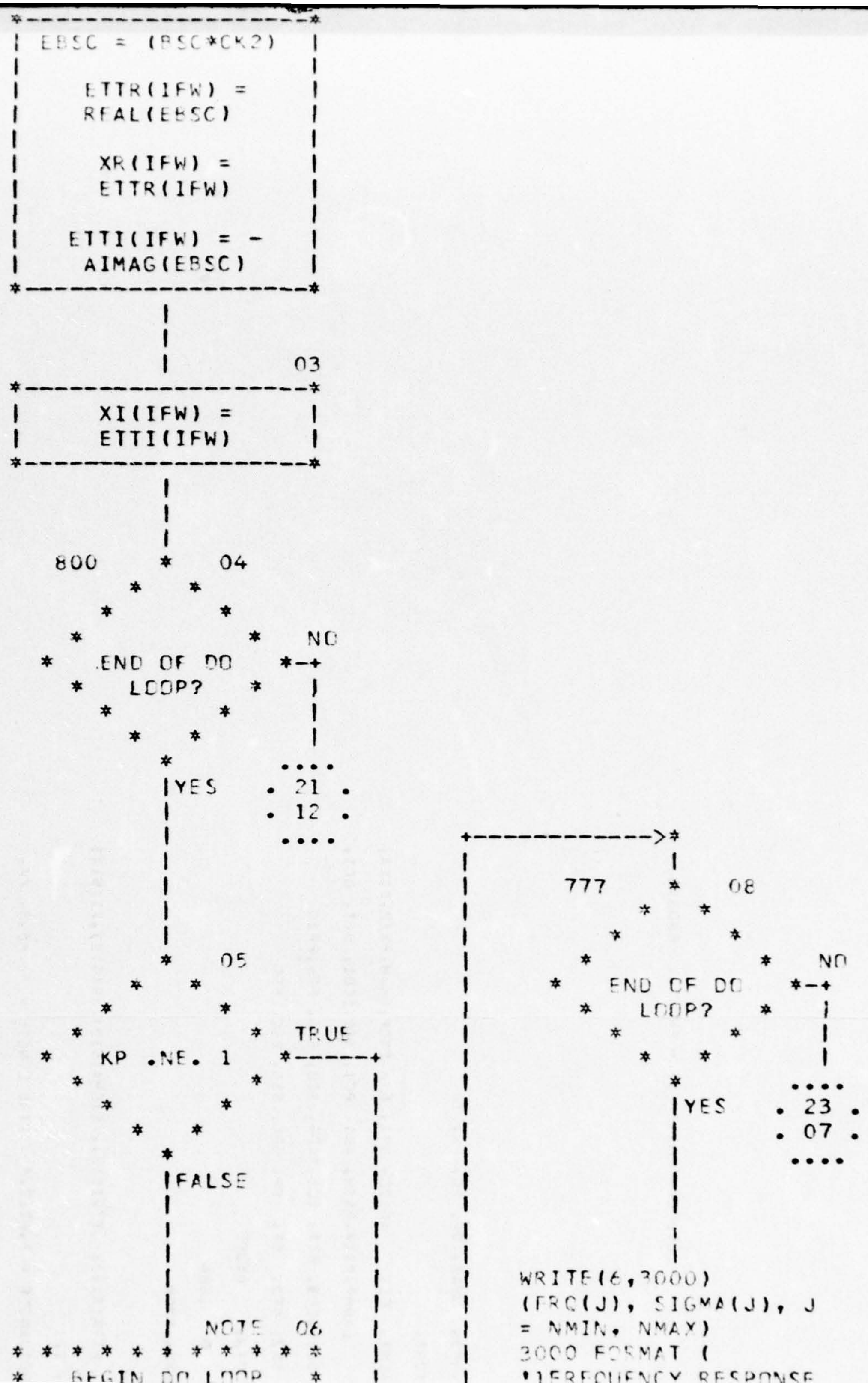

PC1 =
CMPLX(P2,P3)
AC2 =
CMPLX(CTO2F,0.0)
SCD1 =
AC1*PC1*AC2
04
B1 =
1.0/(GKA*CTO2)
B2 = 2.0*ALOG(E1)
BC1 =
CMPLX(B2,PI)
SCD2 =
(-(QC1*AC2*PC1*
PSIP))
05
D1 =
1.0/(GKA*STO2)
D2 = 2.0*ALOG(D1)
DC1 =
CMPLX(D2,PI)
SB4 =
CMPLX(STO2F,0.0)
06
SCD3 =
SB4*(QC1*DC1*
PSIM)
FP5 = XKL*(3.0 +
CT)
FP6 = COS(FP5)
FP7 = SIN(FP5)
FP8 =
CMPLX(FP1,FP2)
11
FP9 =
CMPLX(FP3,FP4)
FP10 =
CMPLX(FP6,FP7)
12
SF1 =
PSIA*PSIA*PSIP*
PSIP*FP8
SF2 =
PSIM*PSIM*FP9
SF3 = -
2.0*PSIP*PSIM*
PSIA*FP10
13
SF = (SF1 + SF2 +
SF3)*CT*EC1
14
D = 1.0 -
(PSIA*PSIA)*RC1
AD = CABS(D)
15
AD .LE.
1.0E-6
TRUE
FALSE
213
17
WRITE TO DEV
6
VIA FORMAT
2005
16
SSS = SF/D
.23.01.
... 221
.23.10.
... 999

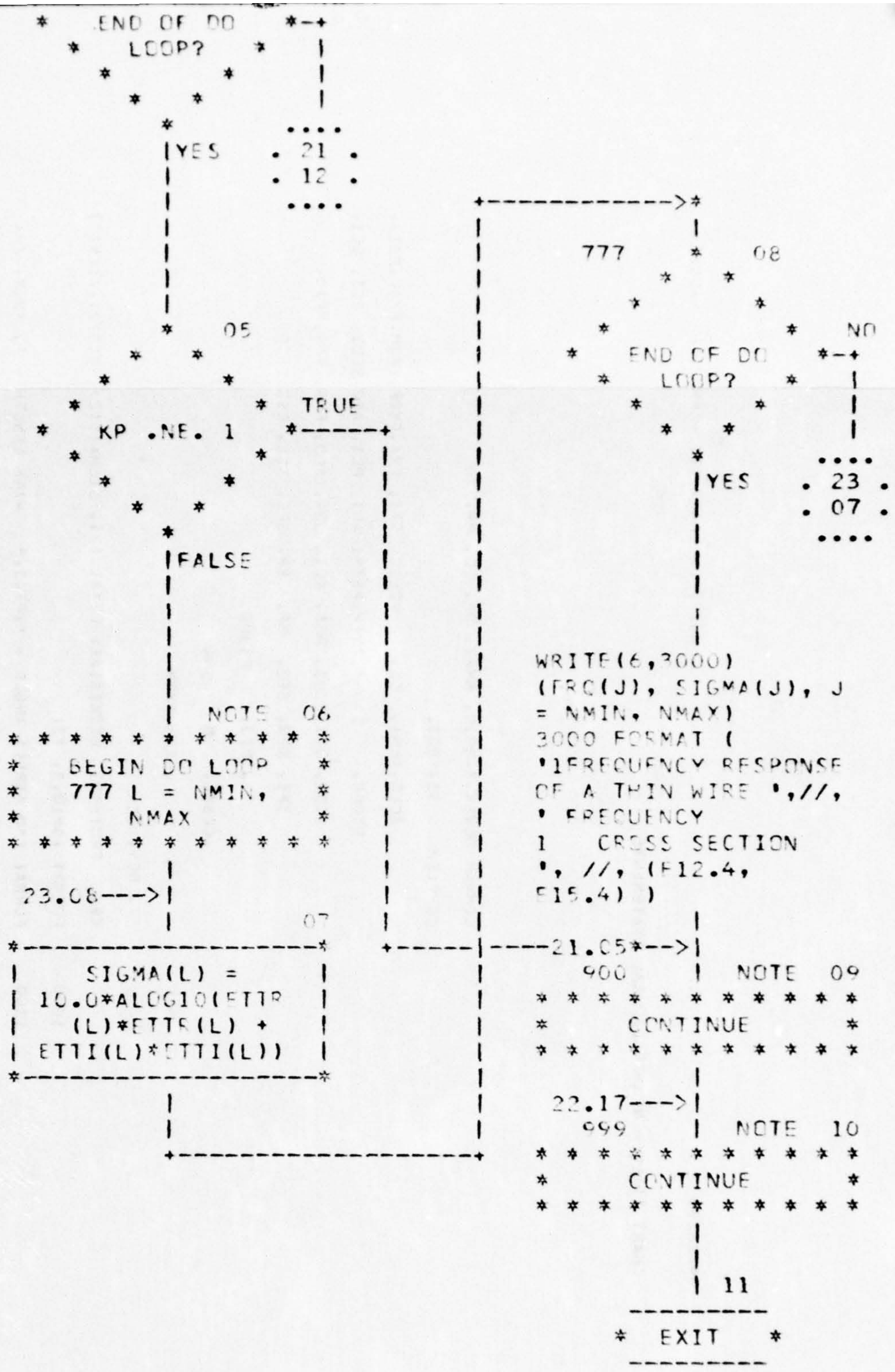
CHART TITLE - SUBROUTINE TARGET(ETTR,ETTI,XR,XI)

```
   * BACKSCATTERED
  FIELD    *

   22.16--->*
     221    |          01
  *--------------------*
  |  ESC = (STI +      |
  |  SCD + SSS)*CS     |
  |                    |
  |        CK1 =       |
  | (4.0*PI)/(XK*XK)   |
  |                    |
  |        CK2 =       |
  | SQRT(CK1)*0.02540  |
  *--------------------*
            |
            |
            |
   **  EBSC =
  SQRT(SIGMA) WITH
  PHASE REFERENCED TO
  FRONT   **
   *                EDGE
  OF    WIRE
   *
            |
            |
            |          02
  *--------------------*
  | EBSC = (BSC*CK2)   |
  |                    |
  |   ETTR(IFW) =      |
  |   REAL(EBSC)       |
  |                    |
  |     XR(IFW) =      |
  |     ETTR(IFW)      |
```

2



EBSC = (BSC*CK2)
ETTR(IFW) = REAL(EBSC)
XR(IFW) = ETTR(IFW)
ETTI(IFW) = - AIMAG(EBSC)
03
XI(IFW) = ETTI(IFW)
800
04
END OF DO LOOP?
NO
YES
21
12
05
KP .NE. 1
TRUE
FALSE
NOTE 06
BEGIN DO LOOP
777
08
END OF DO LOOP?
NO
YES
23
07
WRITE(6,3000) (FRQ(J), SIGMA(J), J = NMIN, NMAX)
3000 FORMAT (
'1FREQUENCY RESPONSE



```
SIGMA(L) =
10.0*ALOG10(ETTR
  (L)*ETTR(L) +
ETTI(L)*ETTI(L))
```

```
WRITE(6,3000)
(FRC(J), SIGMA(J), J
= NMIN, NMAX)
3000 FORMAT (
'1FREQUENCY RESPONSE
OF A THIN WIRE ',//,
' FREQUENCY
1    CROSS SECTION
', //, (F12.4,
E15.4) )
```

L-63

3

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
              COMMON MOVER,M,NMIN, NMAX, DF, FC, PW, TO
              COMPLEX     SS(100),
                         ACLC,ALN1,  CS,     ACL2C, ST1, EXA,EXQP,EXQM,EXAP,PSIA,
                       EXQPP,     EXQMP,PSIP,PSIM, AC1, PC1, AC2,SCD1, BC1, QC1,
                         SCD2, DC1,SCD3, RC1, EC1,SCD4, SCD, FP8, FP9,FP10,
                          SF1, SF2, SF3,  SF,  SP, SPC, SSS, BSC,EBSC
                             , PIJ,    PIJO2
                      ,CC6RL,   D   ,SB4
                             ,EXQA,EXQM
                , RC9, QC9
              REAL  FRQ(512), ETTR(512), ETTI(512), SIGMA(512),XR(512),XI(512)
      1000    FORMAT (3F10.3, I2)
      1100    FORMAT ('0 ASPECT ANGLE = ',F7.2,'   WIRE LENGTH = ', F8.3, //,
                 '  WIRE DIAMETER  = ', F7.4 , // , ' ( LENGTHS ARE IN INCHES )'
              )
      2005    FORMAT ('0 DENOMINATOR IS ZERO , PSIA IS TOO LARGE')
```

```
CHART TITLE - SUFRCUTINF  EXPI(ARCZ,EIXR,EIXI,KP)



                         -----------
                        /   EXPI    /
                         -----------

                      21.17*-->*

                       * THIS SUBROUTINF
                      COMPUTES THE REAL AND
                      IMAGINARY PARTS OF
                      THE
                       *       EXPONENTIAL
                      INTEGRAL E(X) WHERE
                       *       E(X) =
                      INTEGRAL FROM X TO
                      INFINITY OF
                      EXP(IT)/T*DT
                       * REFERENCE -
                      FUNCTIONS OF
                      MATHEMATICAL PHYSICS
                      BY MAGNUS AND
                       *
                      OBERHETTINGER,
                      PP.97-98
                      KP .NE. 0, PRINT OUT
                      REAL AND IMAGINARY
                      PARTS OF EXPONENTIAL
                      INTEGRAL

                               *
                               I         01
                      *-------------------*
                      I  DEL = 0.000001   I
                      I                   I
                      I      GAMMA =      I
                      I    0.57721566     I
                      *-------------------*
                               I
                               I
                               I
                               *   02
                             *   *
                           *       *
```

2

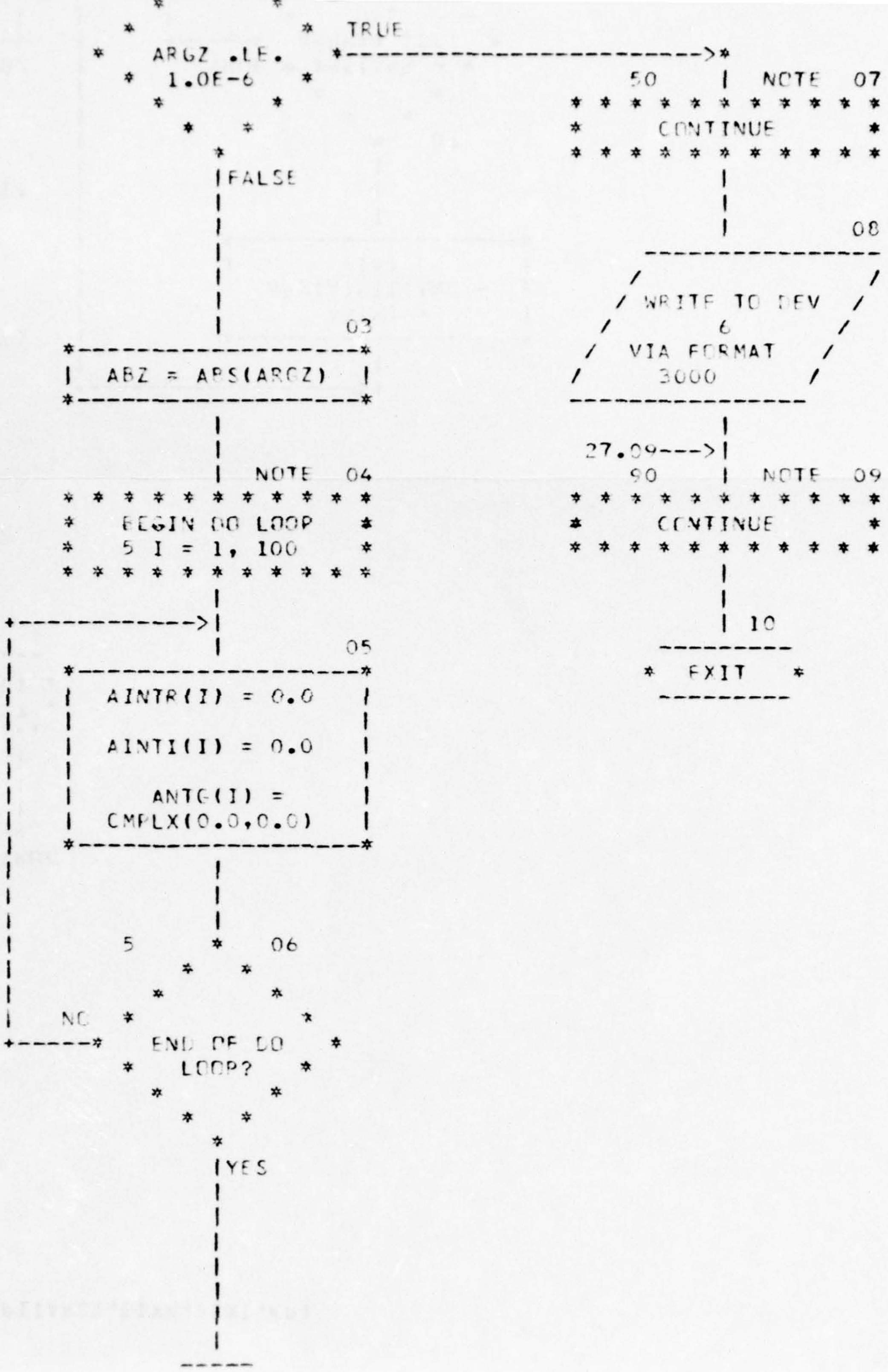
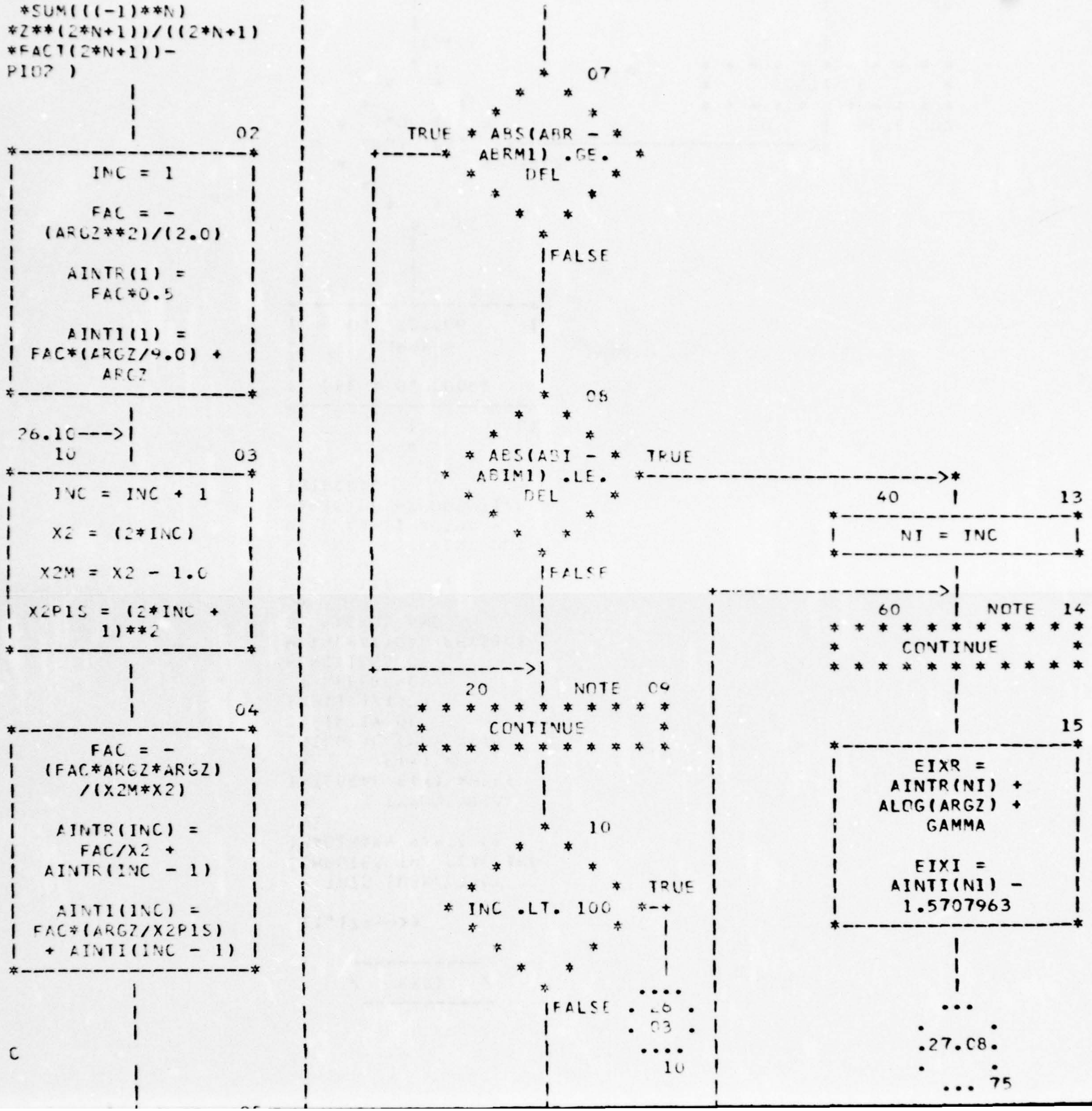

ARGZ .LE. 1.0E-6
TRUE
FALSE
03
ABZ = ABS(ARGZ)
NOTE 04
BEGIN DO LOOP
5 I = 1, 100
05
AINTR(I) = 0.0
AINTI(I) = 0.0
ANTG(I) = CMPLX(0.0,0.0)
5
06
END OF DO LOOP?
NO
YES
50
NOTE 07
CONTINUE
08
WRITE TO DEV 6 VIA FORMAT 3000
27.09--->
90
NOTE 09
CONTINUE
10
EXIT

CHART TITLE - SUBROUTINE EXPI(ARGZ,EIXR,EIXI,KP)

```
 25.06--->*
          |
          *   01
        *   *
      *       *
    *           *  TRUE
  *  ABZ .GE. 15  *--+
    *           *    |
      *       *      |
        *   *        |
          *       ....
          |FALSE . 27 .
          |       . 01 .
          |        ....
          |          9
          |
          |
          |
THIS SERIES USED FOR
0.LT.ABS(X).LT.15
 **  SERIES EXPANSION
INVOLVING CI(X) AND
SI(X)  **
          |
          |             +------------------->*
          |             |                    |      06
E(2) =  GAM   + LN(Z)   |         *-------------------*
+SUM(((-1)**N)          |         |      ABIM1 =      |
*Z**(2*N))/((2*N)       |         |  ABS(AINTI(INC -  |
*FACT(2*N))             |         |        1))        |
+J*(                    |         *-------------------*
 *SUM(((-1)**N)         |                    |
*Z**(2*N+1))/((2*N+1)   |                    |
*FACT(2*N+1))-          |                    |
PIO2 )                  |                    *  07
          |             |                 *   *
```

2


*SUM(((-1)**N)
*Z**(2*N+1))/((2*N+1)
*FACT(2*N+1))-
PIO2)
02
INC = 1
FAC = -
(ARGZ**2)/(2.0)
AINTR(1) =
FAC*0.5
AINTI(1) =
FAC*(ARGZ/9.0) +
ARGZ
26.10--->
10
03
INC = INC + 1
X2 = (2*INC)
X2M = X2 - 1.0
X2P1S = (2*INC +
1)**2
04
FAC = -
(FAC*ARGZ*ARGZ)
/(X2M*X2)
AINTR(INC) =
FAC/X2 +
AINTR(INC - 1)
AINTI(INC) =
FAC*(ARGZ/X2P1S)
+ AINTI(INC - 1)
C
07
TRUE
ABS(ABR -
ABRM1) .GE.
DEL
FALSE
08
ABS(ABI -
ABIM1) .LE.
DEL
TRUE
FALSE
20
NOTE 09
CONTINUE
10
INC .LT. 100
TRUE
FALSE
26
03
10
40
13
NI = INC
60
NOTE 14
CONTINUE
15
EIXR =
AINTR(NI) +
ALOG(ARGZ) +
GAMMA
EIXI =
AINTI(NI) -
1.5707963
27.08
75

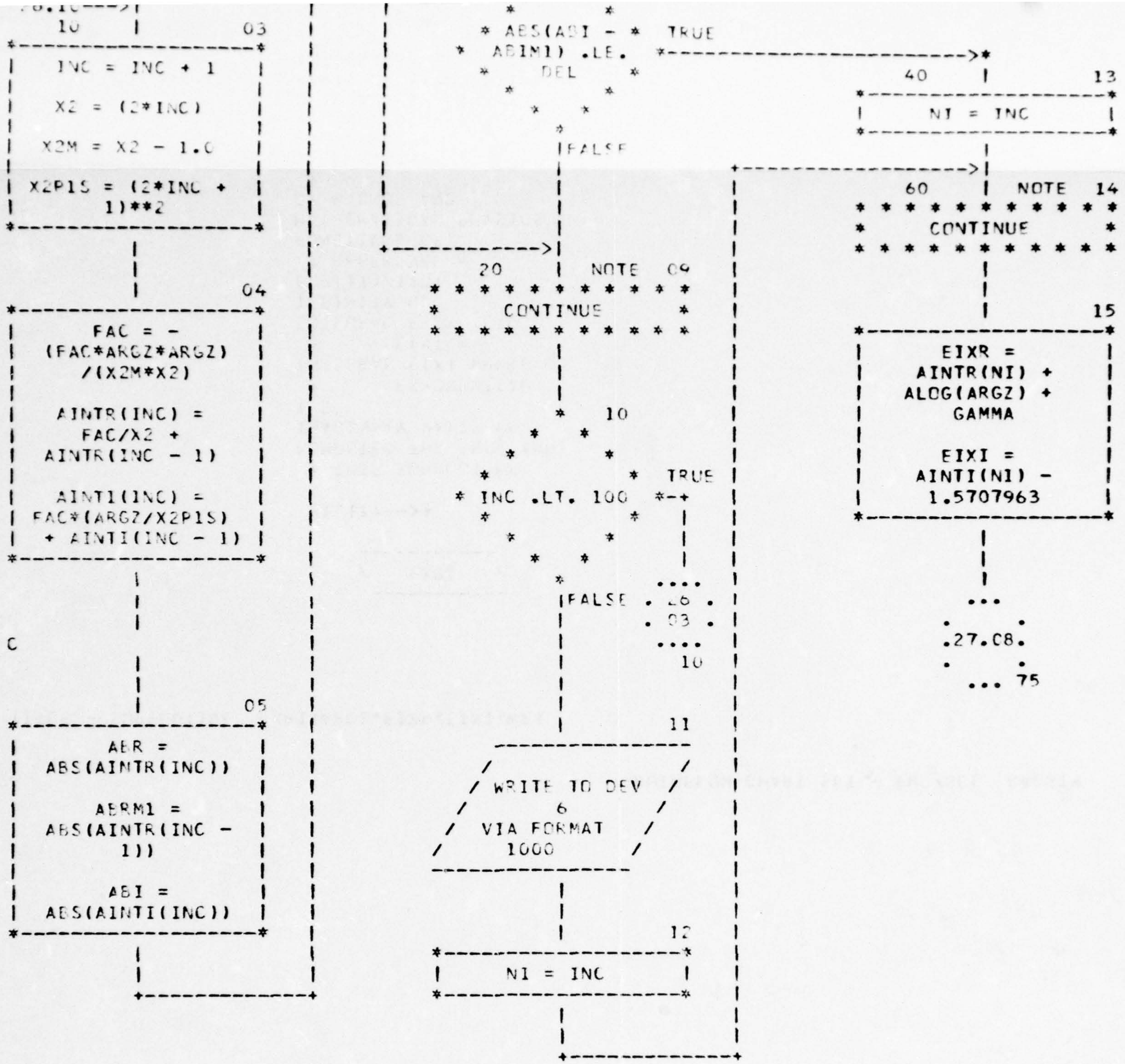
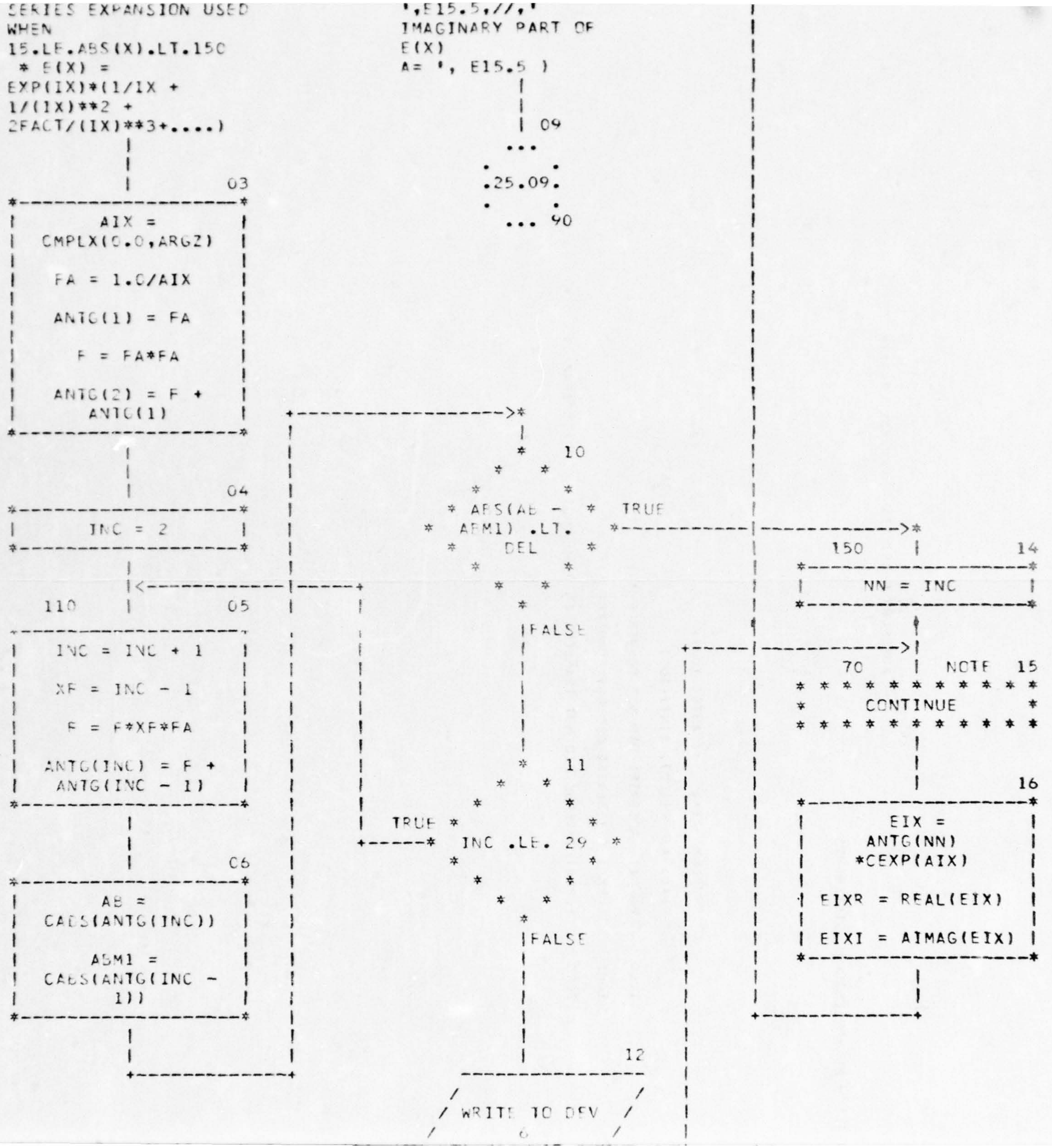

10
03
INC = INC + 1
X2 = (2*INC)
X2M = X2 - 1.0
X2P1S = (2*INC +
1)**2
04
FAC = -
(FAC*ARGZ*ARGZ)
/(X2M*X2)
AINTR(INC) =
FAC/X2 +
AINTR(INC - 1)
AINTI(INC) =
FAC*(ARGZ/X2P1S)
+ AINTI(INC - 1)
C
05
ABR =
ABS(AINTR(INC))
ABRM1 =
ABS(AINTR(INC -
1))
ABI =
ABS(AINTI(INC))
ABS(ABI -
ABIM1) .LE.
DEL
TRUE
FALSE
20
NOTE 09
CONTINUE
10
INC .LT. 100
TRUE
FALSE
26
03
10
11
WRITE TO DEV
6
VIA FORMAT
1000
12
NI = INC
40
13
NI = INC
60
NOTE 14
CONTINUE
15
EIXR =
AINTR(NI) +
ALOG(ARGZ) +
GAMMA
EIXI =
AINTI(NI) -
1.5707963
27.08
75

CHART TITLE - SUBROUTINE EXPI(ARGZ,EIXR,EIXI,KP)

```
C
 **  ASYMPTOTIC
SERIES EXPANSION FOR
INT FROM INF TO X OF
 **
(EXP(-JT)/T)*DT                   FUNCTION SET TO ZERO
                                  IF ARGUMENT
 26.01--->*                       ABS(X).GE.150
    9     |  NOTE  01
* * * * * * * * * * *   +------------------>*
*     CONTINUE      *   |             99    |        07
* * * * * * * * * * *   |         *-------------------*
          |             |         |    EIXR = 0.0     |
          |             |         |                   |
          |             |         |    EIXI = 0.0     |
          *   02        |         *-------------------*
        *   *           |                   |
      *       *         |          26.15*-->|<------------------+
    *           *  TRUE |             75    |  NOTE  08         |
  * ABZ .GE. 150  *-----+         * * * * * * * * * * *         |
    *           *                 *     CONTINUE      *         |
      *       *                   * * * * * * * * * * *         |
        *   *                               |                   |
          *                                 |                   |
          |FALSE                            |                   |
          |                       IF ( KP .EQ. 0) GO TO         |
          |                       90                            |
          |                       WRITE (6,2000) EIXR,          |
          |                       EIXI                          |
          |                       2000 FORMAT ( 'OREAL          |
          |                       PART OF E(X)=                 |
SERIES EXPANSION USED             ',E15.5,//,'                  |
WHEN                              IMAGINARY PART OF             |
15.LE.ABS(X).LT.150               E(X)                          |
 * E(X) =                         A= ', E15.5 )                 |
EXP(IX)*(1/IX +                             |                   |
1/(IX)**2 +                                 |                   |
```

2


SERIES EXPANSION USED
WHEN
15.LE.ABS(X).LT.150
* E(X) =
EXP(IX)*(1/IX +
1/(IX)**2 +
2FACT/(IX)**3+....)
',E15.5,//,'
IMAGINARY PART OF
E(X)
A= ', E15.5)
09
.25.09.
90
03
AIX =
CMPLX(0.0,ARGZ)
FA = 1.0/AIX
ANTG(1) = FA
F = FA*FA
ANTG(2) = F +
ANTG(1)
04
INC = 2
110
05
INC = INC + 1
XF = INC - 1
F = F*XF*FA
ANTG(INC) = F +
ANTG(INC - 1)
06
AB =
CABS(ANTG(INC))
ASM1 =
CABS(ANTG(INC -
1))
10
ABS(AB -
ABM1) .LT.
DEL
TRUE
FALSE
11
INC .LE. 29
TRUE
FALSE
12
WRITE TO DEV
6
150
14
NN = INC
70
NOTE
15
CONTINUE
16
EIX =
ANTG(NN)
*CEXP(AIX)
EIXR = REAL(EIX)
EIXI = AIMAG(EIX)

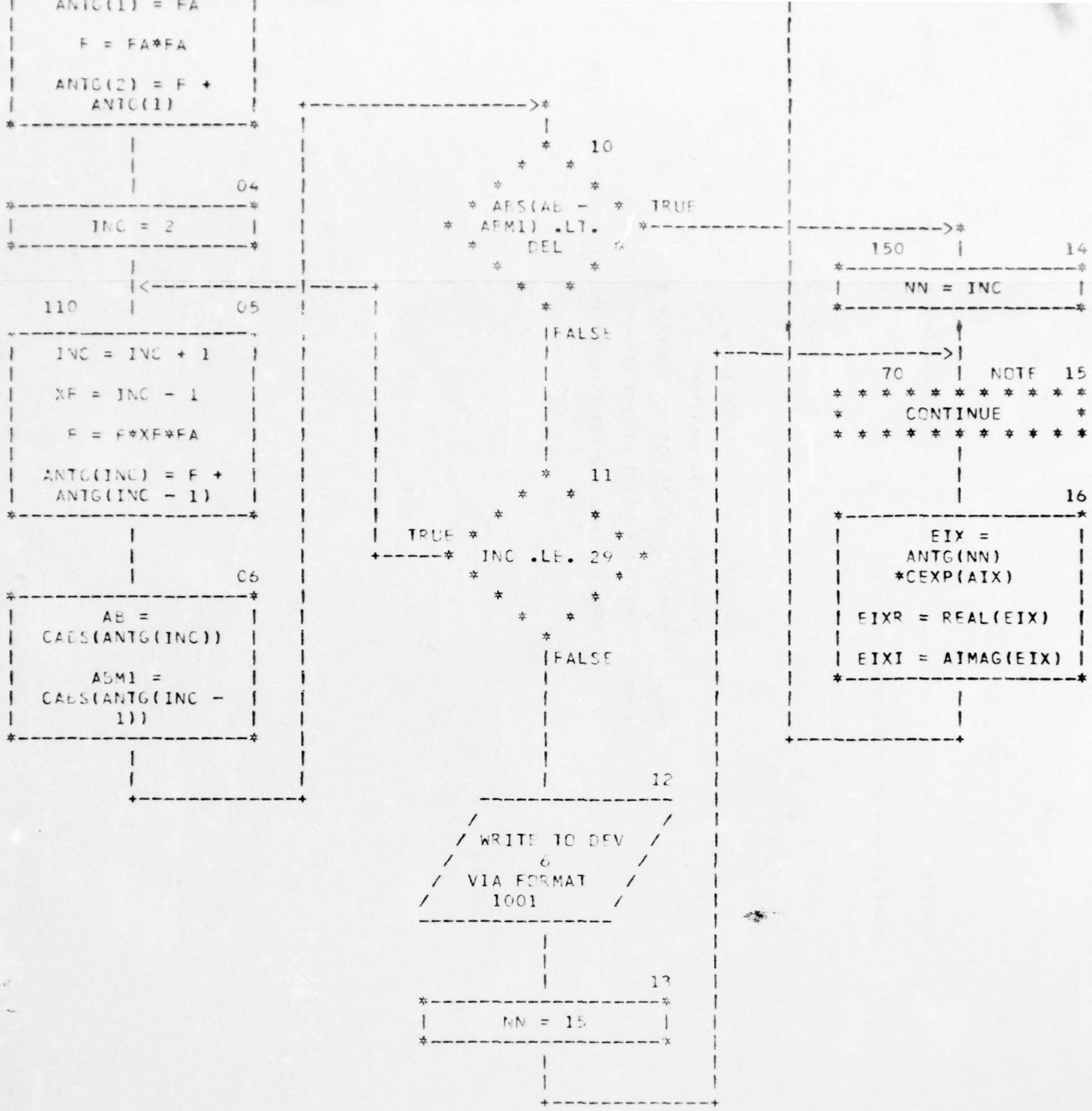

ANTC(1) = FA
F = FA*FA
ANTC(2) = F +
ANTC(1)
04
INC = 2
110
05
INC = INC + 1
XF = INC - 1
F = F*XF*FA
ANTG(INC) = F +
ANTG(INC - 1)
C6
AB =
CABS(ANTG(INC))
ABM1 =
CABS(ANTG(INC -
1))
10
ABS(AB -
ABM1) .LT.
DEL
TRUE
FALSE
11
INC .LE. 29
TRUE
FALSE
12
WRITE TO DEV
6
VIA FORMAT
1001
13
NN = 15
150
14
NN = INC
70
NOTE
15
CONTINUE
16
EIX =
ANTG(NN)
*CEXP(AIX)
EIXR = REAL(EIX)
EIXI = AIMAG(EIX)

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
              COMPLEX  AIX,       ANTG( 100),          FIX,      FA,        F
              REAL  AINTR(100), AINTI(100)
      1000    FORMAT( 'OSERIES DID NOT CONVERGE ' )
      1001    FORMAT ( ' SERIES DID NOT CONVERGE ')
      3000    FORMAT ('OARGZ DID NOT TRANSFER INTO SUBROUTINE OR IS ZERO' )
      SUBROUTINE  TARGET  (ETTR, ETTI,XR,XI)                            RCS6 001
C                                                                       RCS6 002
C     ** THIN WIRE CW RESPONSE   * 0.5 TO 89.5 DEGREES ASPECT ANGLE**   RCS6 003
C     *   SOLUTION BY UFIMTSEV, SIMPLIFIED BY HONG FOR BACKSCATTER   *  RCS6 004
C    * REFERENCE - SHORT PULSE SCATTERING BY A LONG WIRE,  BY S HONG    RCS6 005
C    *           IEEE ON AP, VOL AP-16,NO.3, MAY 1968,PP.338-342        RCS6 006
C                                                                       RCS6 007
C                                                                       RCS6 008
      COMMON MOVER,M,NMIN, NMAX, DF, FC, PW, TO                         RCS6 009
C     NMIN = MINIMUM FREQUENCY SAMPLE                                   RCS6 010
C     NMAX = MAXIMUM FREQUENCY SAMPLE                                   RCS6 011
C     DF   = FREQUENCY INCREMENT IN MHZ                                 RCS6 012
C     FC   = CARRIER FREQUENCY IN GHZ                                   RCS6 013
C                                                                       RCS6 014
      COMPLEX    SS(100),                                               RCS6 015
     A       ACLC,ALN1,  CS,     ACL2C, ST1, EXA,EXQP,EXQM,EXAP,PSIA,   RCS6 016
     B      EXQPP,     EXQMP,PSIP,PSIM, AC1, PC1, AC2,SCD1, BC1, QC1,   RCS6 017
     C       SCD2, DC1,SCD3, RC1, EC1,SCD4, SCD, FP8, FP9,FP10,         RCS6 018
     D        SF1, SF2, SF3,  SF,  SP, SPC, SSS, BSC,EBSC               RCS6 019
     E            , PIJ,    PIJO2                                       RCS6 020
     F      ,CC6RL,   D   ,SB4                                          RCS6 021
     G            ,EXQA,EXQM                                            RCS6 022
     1 , RC9, QC9                                                       RCS6 023
      REAL  FRQ(512), ETTR(512), ETTI(512), SIGMA(512),XR(512),XI(512)  RCS6 024
C                                                                       RCS6 025
```

```
   693                  I   , RC9, CC9                                                        RCS6 022
                                                                                              RCS6 023
   694                      REAL  FRQ(512), ETTR(512), ETTI(512), SIGMA(512),XR(512),XI(512)  RCS6 024
   695             C                                                                          RCS6 025
```

2

```
04/26/76        INPUT LISTING                               AUTOFLOW CHART SET - FWD/SCL  RADSIM

  CARD NO         ****                               CONTENTS                                  ****

   696                  READ(5,1000) THETA, XL, DIA, KP                                       RCS6 026
   697            C     KP = 1, PRINT OUT RCS VERSUS FREQUENCY                                RCS6 027
   698             1000 FORMAT (3F10.3, I2)                                                   RCS6 028
   699                  WRITE(6,1100) THETA, XL, DIA                                          RCS6 030
   700             1100 FORMAT ('0 ASPECT ANGLE = ',F7.2,'   WIRE LENGTH = ', F8.3, //,       RCS6 031
   701                 A    ' WIRE DIAMETER  = ', F7.4 , // , ' ( LENGTHS ARE IN INCHES )'RCS6 032
   702                 B)                                                                     RCS6 033
   703                  IF (THETA .LT. 0.5  .OR. THETA .GT. 89.5 ) GO TO 900                  RCS6 034
   704                  RAD = DIA/2.0                                                         RCS6 035
   705                  PI = 3.1415926                                                        RCS6 036
   706                  PIJ = CMPLX(0.0,PI)                                                   RCS6 037
   707                  PIO2= PI/2.0                                                          RCS6 038
   708                  PIJO2 = CMPLX(0.0,PIO2)                                               RCS6 039
   709                  GAM = 1.781072                                                        RCS6 040
   710                  TH  = THETA * (180.0 / PI)                                            RCS6 041
   711                  CT = COS(TH)                                                          RCS6 042
```

```
C      KP = 1, PRINT OUT RCS VERSUS FREQUENCY                          RCS6 027
  1000 FORMAT (3F10.3, I2)                                             RCS6 028
       WRITE(6,1100) THETA, XL, DIA                                    RCS6 030
  1100 FORMAT ('0 ASPECT ANGLE = ',F7.2,'   WIRE LENGTH = ', F8.3, //,   RCS6 031
     A   ' WIRE DIAMETER  = ', F7.4 , // , ' ( LENGTHS ARE IN INCHES )'RCS6 032
     B)                                                                RCS6 033
       IF (THETA .LT. 0.5  .OR. THETA .GT. 89.5 ) GO TO 900            RCS6 034
       RAD = DIA/2.0                                                   RCS6 035
       PI = 3.1415926                                                  RCS6 036
       PIJ = CMPLX(0.0,PI)                                             RCS6 037
       PIO2= PI/2.0                                                    RCS6 038
       PIJO2 = CMPLX(0.0,PIO2)                                         RCS6 039
       GAM = 1.781072                                                  RCS6 040
       TH  = THETA * (180.0 / PI)                                      RCS6 041
       CT = COS(TH)                                                    RCS6 042
       CTO2 = COS(TH/2.0)                                              RCS6 043
       CTO2S= CTO2 * CTO2                                              RCS6 044
       CTO2F= CTO2S * CTO2S                                            RCS6 045
       ST = SIN (TH)                                                   RCS6 C46
       ST1 = SIN(TH*2.0)                                               RCS6 047
       STO2 = SIN(TH/2.0)                                              RCS6 048
       STO2S = STO2 * STO2                                             RCS6 049
       STO2F = STO2S * STO2S                                           RCS6 050
       OPCT = 1.0+ CT                                                  RCS6 051
       OMCT = 1.0- CT                                                  RCS6 052
       AKTH = 2.0/(ST*ST1)                                             RCS6 053
C                                                                      RCS6 054
```

3

L-69a

```
      DO 300  IFW = NMIN,NMAX                                          RCS6 055
      XI= IFW - 1                                                      RCS6 056
      FREQ    = XI * DF / 1000.0                                       RCS6 057
      FRQ( IFW) = FREQ                                                 RCS6 058
      XK =  ( .53234454 * FREQ )                                       RCS6 059
      XKL =  XK* XL                                                    RCS6 060
      XKA =  XK* RAD                                                   RCS6 061
      GKA =  GAM* XKA                                                  RCS6 062
      C1 = 2.0/( GKA * ST)                                             RCS6 063
      ACL = ALOG ( C1 )                                                RCS6 064
      ACLC =  CMPLX ( ACL, PIO2 )                                      RCS6 065
      CS =((0.0, 1.0)/(ACLC*ACLC))*(-AKTH)                             RCS6 066
C                                                                      RCS6 067
C     ** DETERMINATION OF INTEGRAL AND OTHER TERMS USED IN MULTIPLE ** RCS6 068
C        ORDER SCATTERING                                              RCS6 069
C                                                                      RCS6 070
      C5L = ALOG( GKA)                                                 RCS6 071
      C6R = (2.0/XKA)*(XKL/GKA)                                        RCS6 072
      C6RL= ALOG( C6R)                                                 RCS6 073
      CC6RL = CMPLX(C6RL,PIO2)                                         RCS6 074
      EAA = XKL*2.0                                                    RCS6 075
      CALL EXPI ( EAA, EXRA, EXIA, 0 )                                 RCS6 076
      EXWA = CMPLX( EXRA, EXIA)                                        RCS6 077
      E1C= COS( EAA)                                                   RCS6 078
      E1S= SIN( EAA)                                                   RCS6 079
```

```
    747          E1C= COS( EAA)                                                   RCS6 078
    748          E1S= SIN( EAA)                                                   RCS6 079
    749          RC1 = CMPLX (E1C, E1S)                                           RCS6 080
    750          RC9     = CMPLX( E1C, -E1S )                                     RCS6 081
    751          PSIA    = (PIJ - 2.0*C5L) / (CC6RL - EXQA*RC9)                   RCS6 082
    752      C                                                                    RCS6 083
    753          EQPA = XKL*OMCT                                                  RCS6 084




 04/26/76    INPUT LISTING                       AUTOFLOW CHART SET - FWD/SCL  RADSIM

  CARD NO       ****                        CONTENTS                                ****

    754          CALL EXPI (EQPA, EXR, EXI, 0 )                                   RCS6 085
    755          EXOP= CMPLX( EXR, EXI )                                          RCS6 086
    756          E2C =COS( EQPA)                                                  RCS6 087
    757          E2S =SIN( EQPA)                                                  RCS6 088
    758          EXQPP = CMPLX( E2C,-E2S)                                         RCS6 089
    759          C7P = GKA*GKA*OMCT/2.0                                           RCS6 090
    760          C7PL = ALOG(C7P)                                                 RCS6 091
    761          PSIP = (PIJ - C7PL)/(CC6RL - EXQP*EXQPP)                         RCS6 092
    762      C                                                                    RCS6 093
    763          EQMA = XKL*OPCT                                                  RCS6 094
    764          CALL EXPI ( EQMA, EXRM, EXIM, 0 )                                RCS6 095
    765          EXQM = CMPLX( EXRM, EXIM)                                        RCS6 096
    766          C2      = COS(EQMA)                                              RCS6 097
```

```
      PSIP = (PIJ - C7PL)/(CC6RL - EXQP*EXQPP)                   RCS6 092
C                                                                RCS6 093
      EQMA = XKL*OPCT                                            RCS6 094
      CALL EXPI ( EQMA, EXRM, EXIM, 0 )                          RCS6 095
      EXQM = CMPLX( EXRM, EXIM)                                  RCS6 096
      Q2      = COS(EQMA)                                        RCS6 097
      Q3      = SIN(EQMA)                                        RCS6 098
      QC1 = CMPLX( Q2, Q3)                                       RCS6 099
      QC9     = CMPLX( Q2, -Q3 )                                 RCS6 100
      C7M = GKA*GKA*OPCT/2.0                                     RCS6 101
      C7ML = ALOG(C7M)                                           RCS6 102
      PSIM    = (PIJ - C7ML) / (CC6RL - EXQM*QC9)                RCS6 103
C                                                                RCS6 104
C     * UTIL1ZATION OF FACTORS IN SCATTERED FIELD EXPRESSION *   RCS6 105
C                                                                RCS6 106
C     **  FIRST ORDER SCATTERING   **                            RCS6 107
C                                                                RCS6 108
      C2 = 1.0 / (GKA * STO2S)                                   RCS6 109
      ACL2 = ALOG (C2)                                           RCS6 110
      ACL2C = CMPLX ( ACL2, PIO2 )                               RCS6 111
      ST1 = -STO2F * ACL2C                                       RCS6 112
```

3

```
C                                                                 RCS6 113
C     *   RETURN FROM TRAILING EDGE OF WIRE    *                  RCS6 114
C                                                                 RCS6 115
      A1    = 1.0/(GKA* CTO2S)                                    RCS6 116
      A2    = ALOG(A1)                                            RCS6 117
      AC1   = CMPLX( A2,PIO2)                                     RCS6 118
      P1 =  2.0*XKL*CT                                            RCS6 119
      P2 = COS(P1)                                                RCS6 120
      P3 = SIN(P1)                                                RCS6 121
      PC1= CMPLX( P2, P3)                                         RCS6 122
      AC2 = CMPLX( CTO2F,0.0)                                     RCS6 123
      SCD1 = AC1*PC1*AC2                                          RCS6 124
C                                                                 RCS6 125
      B1 = 1.0/(GKA*CTO2)                                         RCS6 126
      B2 = 2.0*ALOG(B1)                                           RCS6 127
      BC1 = CMPLX( B2,PI)                                         RCS6 128
      SCD2 =(-(QC1*AC2*BC1*PSIP))                                 RCS6 129
C                                                                 RCS6 130
      D1 = 1.0/(GKA*STO2)                                         RCS6 131
      D2 = 2.0*ALOG(D1)                                           RCS6 132
      DC1= CMPLX(D2,PI)                                           RCS6 133
      SB4      = CMPLX( STO2F, 0.0)                               RCS6 134
      SCD3 = SB4  *(QC1*DC1*PSIM)                                 RCS6 135
C                                                                 RCS6 136
      E1 = 1.0/GKA                                                RCS6 137
```

```
 805        C                                                                   RCS6 136
 806                  E1 = 1.0/GKA                                              RCS6 137
 807                  E2 = ALOG( E1)                                            RCS6 138
 808                  EC1= CMPLX( E2,PIO2)                                      RCS6 139
 809                  SCD4 = CT*(RC1*EC1*PSIP*PSIP)                             RCS6 140
 810        C                                                                   RCS6 141
 811                  SCD = SCD1 + SCD2 + SCD3 + SCD4                           RCS6 142


04/26/76       INPUT LISTING                      AUTOFLOW CHART SET - FWD/SCL  RADSIM

CARD NO     ****                      CONTENTS                                  ****

 812        C                                                                   RCS6 143
 813                  FP1 = COS(4.0*XKL)                                        RCS6 144
 814                  FP2 = SIN(4.0*XKL)                                        RCS6 145
 815                  FP3 = COS(2.0*XKL*OPCT)                                   RCS6 146
 816                  FP4 = SIN(2.0*XKL*OPCT)                                   RCS6 147
 817                  FP5 = XKL*(3.0+CT)                                        RCS6 148
 818                  FP6 = COS(FP5)                                            RCS6 149
 819                  FP7 = SIN(FP5)                                            RCS6 150
 820                  FP8  = CMPLX( FP1,FP2)                                    RCS6 151
 821                  FP9  = CMPLX( FP3,FP4)                                    RCS6 152
 822                  FP10 = CMPLX( FP6,FP7)                                    RCS6 153
 823        C                                                                   RCS6 154
 824                  SF1 = PSIA*PSIA*PSIP*PSIP*FP8                             RCS6 155
 825                  SF2 = PSIM*PSIM*FP9                                       RCS6 156
```

```
      FP1 = COS(4.0*XKL)                                          RCS6 144
      FP2 = SIN(4.0*XKL)                                          RCS6 145
      FP3 = COS(2.0*XKL*OPCT)                                     RCS6 146
      FP4 = SIN(2.0*XKL*OPCT)                                     RCS6 147
      FP5 = XKL*(3.0+CT)                                          RCS6 148
      FP6 = COS(FP5)                                              RCS6 149
      FP7 = SIN(FP5)                                              RCS6 150
      FP8  = CMPLX( FP1,FP2)                                      RCS6 151
      FP9  = CMPLX( FP3,FP4)                                      RCS6 152
      FP10 = CMPLX( FP6,FP7)                                      RCS6 153
C                                                                 RCS6 154
      SF1 = PSIA*PSIA*PSIP*PSIP*FP8                               RCS6 155
      SF2 = PSIM*PSIM*FP9                                         RCS6 156
      SF3 = -2.0*PSIP*PSIM*PSIA*FP10                              RCS6 157
      SF = (SF1+SF2+SF3)*CT*EC1                                   RCS6 158
C                                                                 RCS6 159
      D =1.0-(PSIA*PSIA)* RC1                                     RCS6 160
      AD =CABS(D)                                                 RCS6 161
      IF (AD .LE. 1.0E-6 ) GO TO 213                              RCS6 162
      SSS =  SF/D                                                 RCS6 163
      GO TO 221                                                   RCS6 164
  213 WRITE (6, 2005)                                             RCS6 165
 2005 FORMAT ('0 DENOMINATOR IS ZERO , PSIA IS TOO LARGE')        RCS6 166
      GO TO 999                                                   RCS6 169
C                                                                 RCS6 170
C     * BACKSCATTERED FIELD   *                                   RCS6 171
C                                                                 RCS6 172
```

3

L-696

```
  221 EBSC = (ST1 + SCD + SSS)* CS                                     RCS6 173
      CK1 = (4.0* PI)/(XK*XK)                                          RCS6 174
      CK2 = SQRT(CK1)*0.02540                                          RCS6 175
C                                                                      RCS6 176
C     **  EBSC = SQRT(SIGMA) WITH PHASE REFERENCED TO FRONT  **        RCS6 177
C      *               EDGE   OF   WIRE                      *        RCS6 178
      EBSC =(EBSC* CK2)                                                RCS6 179
      ETTR(IFW)= REAL(EBSC)                                            RCS6 180
       XR(IFW) = ETTR(IFW)
      ETTI(IFW)=-AIMAG(EBSC)                                           RCS6 181
       XI(IFW) = ETTI(IFW)
  800 CONTINUE                                                         RCS6 182
C                                                                      RCS6 183
      IF (KP .NE. 1)  GO TO 900                                        RCS6 184
      DO 777  L = NMIN, NMAX                                           RCS6 185
      SIGMA(L)  = 10.0 * ALOG10(ETTR(L)*ETTR(L) + ETTI(L)*ETTI(L))     RCS6 186
  777 CONTINUE                                                         RCS6 187
C     WRITE(6,3000) (FRQ(J), SIGMA(J), J = NMIN, NMAX)                 RCS6 188
C3000 FORMAT ( '1FREQUENCY RESPONSE OF A THIN WIRE ',//,  ' FREQUENCY   RCS6 189
C    1   CROSS SECTION     ', //, (F12.4,  E15.4) )                    RCS6 190
  900 CONTINUE                                                         RCS6 191
  999 CONTINUE                                                         RCS6 192
      RETURN                                                           RCS6 193
      END                                                              RCS6 194
```

```
864              SUBROUTINE EXPI ( ARGZ, EIXR, EIXI, KP)                             RCS6 195
865         C                                                                        RCS6 196
866         C   * THIS SUBROUTINE COMPUTES THE REAL AND IMAGINARY PARTS OF THE       RCS6 197
867         C   *      EXPONENTIAL INTEGRAL E(X) WHERE                               RCS6 198
868         C   *      E(X) =  INTEGRAL FROM X TO INFINITY OF EXP(IT)/T*DT           RCS6 199
869         C   * REFERENCE - FUNCTIONS OF MATHEMATICAL PHYSICS BY MAGNUS AND        RCS6 200

04/26/76      INPUT LISTING                         AUTOFLOW CHART SET - FWD/SCL  RADSIM

CARD NO       ****                        CONTENTS                                    ****

870         C   *         OBERHETTINGER, PP.97-98                                    RCS6 201
871         C      KP .NE. 0, PRINT OUT REAL AND IMAGINARY PARTS OF EXPONENTIAL      RCS6 202
872         C            INTEGRAL                                                    RCS6 203
873              COMPLEX  AIX,      ANTG( 100),         EIX,      FA,       F        RCS6 204
874              REAL  AINTR(100), AINTI(100)                                        RCS6 205
875         C                                                                        RCS6 206
876              DEL = 0.000001                                                      RCS6 207
877              GAMMA = 0.57721566                                                  RCS6 208
878              IF ( ARGZ .LE. 1.0E-6) GO TO 50                                     RCS6 209
879              ABZ = ABS(ARGZ)                                                     RCS6 210
880              DO 5  I=1,100                                                       RCS6 211
```

```
C     KP .NE. 0, PRINT OUT REAL AND IMAGINARY PARTS OF EXPONENTIAL      RCS6 202
C              INTEGRAL                                                  RCS6 203
      COMPLEX  AIX,      ANTG( 100),         EIX,      FA,        F     RCS6 204
      REAL  AINTR(100), AINTI(100)                                       RCS6 205
C                                                                        RCS6 206
      DEL = 0.000001                                                     RCS6 207
      GAMMA = 0.57721566                                                 RCS6 208
      IF ( ARGZ .LE. 1.0E-6) GO TO 50                                    RCS6 209
      ABZ = ABS(ARGZ)                                                    RCS6 210
      DO 5  I=1,100                                                      RCS6 211
      AINTR(I)= 0.0                                                      RCS6 212
      AINTI(I)= 0.0                                                      RCS6 213
      ANTG(I) = CMPLX(0.0,0.0)                                           RCS6 214
    5 CONTINUE                                                           RCS6 215
      IF (  ABZ .GE. 15) GO TO 9                                         RCS6 216
C                                                                        RCS6 217
C     THIS SERIES USED FOR  0.LT.ABS(X).LT.15                            RCS6 218
C     **  SERIES EXPANSION INVOLVING CI(X) AND SI(X)   **                RCS6 219
C                                                                        RCS6 220
C     E(Z) =   GAM    + LN(Z) +SUM(((-1)**N)*Z**(2*N))/((2*N)*FACT(2*N)) RCS6 221
C          +J*(     *SUM(((-1)**N)*Z**(2*N+1))/((2*N+1)*FACT(2*N+1))-   RCS6 222
C          PIO2 )                                                        RCS6 223
C                                                                        RCS6 224
      INC =1                                                             RCS6 225
      FAC = -(ARGZ**2)/(2.0)                                             RCS6 226
      AINTR(1) =  FAC * 0.5                                              RCS6 227
      AINTI(1) = FAC*(ARGZ/4.0) + ARGZ                                   RCS6 228
```

2

L-69d

L-69e

```
      C                                                         RCS6 229
   10 INC = INC + 1                                             RCS6 230
      X2 =(2* INC)                                              RCS6 231
      X2M = X2 -1.0                                             RCS6 232
      X2P1S = (2*INC + 1 )**2                                   RCS6 233
      FAC = -(FAC*ARGZ*ARGZ)/(X2M*X2)                           RCS6 234
      FAC = -(FAC*ARGZ*ARGZ)/(X2M*X2)                           RCS6 234
      AINTR(INC) = FAC/X2 + AINTR(INC-1)                        RCS6 235
      AINTI(INC) = FAC*(ARGZ/X2P1S) +AINTI(INC-1)               RCS6 236
      CC                                                        RCS6 237
      ABR  = ABS( AINTR(INC))                                   RCS6 238
      ABRM1= ABS( AINTR(INC-1))                                 RCS6 239
      ABI  = ABS( AINTI(INC)  )                                 RCS6 240
      ABIM1= ABS( AINTI(INC-1))                                 RCS6 241
      IF ( ABS( ABR - ABRM1) .GE. DEL) GO TO 20                 RCS6 242
      IF ( ABS( ABI - ABIM1) .LE. DEL) GO TO 40                 RCS6 243
   20 CONTINUE                                                  RCS6 244
      C                                                         RCS6 245
      IF ( INC .LT. 100 ) GO TO 10                              RCS6 246
      WRITE (6, 1000)                                           RCS6 247
 1000 FORMAT( 'OSERIES DID NOT CONVERGE ' )                     RCS6 248
      NI = INC                                                  RCS6 249
      GO TO 60                                                  RCS6 250
   40 NI= INC                                                   RCS6 251
   60 CONTINUE                                                  RCS6 252
      FIXP = AINTR(NI) + ALOG(ARGZ) + GAMMA                     RCS6 253
   40 NI= INC                                                          RCS6 251
   60 CONTINUE                                                         RCS6 252
      EIXR = AINTR(NI) + ALOG(ARGZ) + GAMMA                            RCS6 253
      EIXI = AINTI(NI) - 1.5707963                                     RCS6 254
      GO TO 75                                                         RCS6 255
CC                                                                     RCS6 256
C     **  ASYMPTOTIC SERIES EXPANSION FOR INT FROM INF TO X OF  **     RCS6 257
C          (EXP(-JT)/T)*DT                                             RCS6 258
C                                                                      RCS6 259
    9 CONTINUE                                                         RCS6 260
      IF (  ABZ .GE. 150) GO TO 99                                     RCS6 261
C     SERIES EXPANSION USED WHEN 15.LE.ABS(X).LT.150                   RCS6 262
C    * E(X) = EXP(IX)*(1/IX + 1/(IX)**2 + 2FACT/(IX)**3+....)          RCS6 263
C                                                                      RCS6 264
      AIX =  CMPLX(0.0,ARGZ )                                          RCS6 265
      FA = 1.0/AIX                                                     RCS6 266
      ANTG(1) = FA                                                     RCS6 267
      F   = FA*FA                                                      RCS6 268
      ANTG(2) = F      + ANTG(1)                                       RCS6 269
      INC = 2                                                          RCS6 270
C                                                                      RCS6 271
  110 INC = INC + 1                                                    RCS6 272
      XF = INC-1                                                       RCS6 273
      F = F*XF*FA                                                      RCS6 274
      ANTG(INC) = F + ANTG(INC-1)                                      RCS6 275
      AE =  CABS (ANTG(INC))                                           RCS6 276
      ABM1 =CABS (ANTG(INC-1))                                         RCS6 277
      IF ( ABS(AB - ABM1) .LT. DEL) GO TO150                           RCS6 278
      IF (  ABZ .GE. 150) GO TO 99                                     RCS6 261
C     SERIES EXPANSION USED WHEN 15.LE.ABS(X).LT.150                   RCS6 262
C   * E(X) = EXP(IX)*(1/IX + 1/(IX)**2 + 2FACT/(IX)**3+....)           RCS6 263
C                                                                      RCS6 264
      AIX =  CMPLX(0.0,ARGZ )                                          RCS6 265
      FA = 1.0/AIX                                                     RCS6 266
      ANTG(1) = FA                                                     RCS6 267
      F   = FA*FA                                                      RCS6 268
      ANTG(2) = F      + ANTG(1)                                       RCS6 269
      INC = 2                                                          RCS6 270
C                                                                      RCS6 271
  110 INC = INC + 1                                                    RCS6 272
      XF = INC-1                                                       RCS6 273
      F = F*XF*FA                                                      RCS6 274
      ANTG(INC) = F + ANTG(INC-1)                                      RCS6 275
      AB =  CABS (ANTG(INC))                                           RCS6 276
      ABM1 =CABS (ANTG(INC-1))                                         RCS6 277
      IF ( ABS(AB - ABM1) .LT. DEL) GO TO150                           RCS6 278
      IF ( INC .LE. 29) GO TO110                                       RCS6 279
C                                                                      RCS6 280
      WRITE ( 6, 1001 )                                                RCS6 281
 1001 FORMAT ( ' SERIES DID NOT CONVERGE ')                            RCS6 282
      NN =  15                                                         RCS6 283
      GO TO 70                                                         RCS6 284
  150 NN = INC                                                         RCS6 285
   70 CONTINUE                                                         RCS6 286
```

```
C                                                                        RCS6 287
      EIX = ANTG( NN )*CEXP(AIX)                                         RCS6 288
      EIXR = REAL(EIX)                                                   RCS6 289
      EIXI = AIMAG(EIX)                                                  RCS6 290
      GO TO 75                                                           RCS6 291
C                                                                        RCS6 292
C     FUNCTION SET TO ZERO IF ARGUMENT ABS(X).GE.150                     RCS6 293
   99 EIXR = 0.0                                                         RCS6 294
      EIXI = 0.0                                                         RCS6 295
   75 CONTINUE                                                           RCS6 296
C                                                                        RCS6 297
C     IF ( KP .EQ. 0) GO TO 90                                           RCS6 298
C     WRITE (6,2000) EIXR, EIXI                                          RCS6 299
C2000 FORMAT ( '0REAL PART OF E(X)= ',E15.5,//,' IMAGINARY PART OF E(X) RCS6 300
C    A= ', E15.5 )                                                       RCS6 301
      GO TO 90                                                           RCS6 302
   50 CONTINUE                                                           RCS6 303
      WRITE (6, 3000)                                                    RCS6 304
 3000 FORMAT ('0ARGZ DID NOT TRANSFER INTO SUBROUTINE OR IS ZERO' )      RCS6 305
   90 CONTINUE                                                           RCS6 306
      RETURN                                                             RCS6 307
      END                                                                RCS6 308
```

L.5 FRUSTRUM - CYLINDER - FRUSTRUM

The far-field scattering from a frustrum-cylinder-frustrum target configuration shown in Figure L.5-1 has been formulated using the Ruck-Ufimtsev technique (Ref. 6).

The expressions of the target frequency response are the following:

$$e(\theta)_{\begin{Bmatrix} V \\ H \end{Bmatrix}} = \mp \sqrt{\pi} \Big\{ g(1) + g(2) + g(3) + g(4) + g(5) + g(6) + g(7) + g(8) \Big\}$$

where (1) the g(m) are the sum of the scattering due to the uniform and non-uniform current associated with edge m

(2) The upper and lower signs correspond to vertical and horizontal polarization, respectively,

(3) e^{-iwt} harmonic time variation is assumed, and

(4) the scattering geometry is presented in Figure 5-1

For $0 < \theta < \pi/2$,

$$g(1) = a_1 e^{ip_1} \Big\{ JJ_{11} - \Big[C(n_1) \mp B(n_1, \pi/2+\theta) \pm 0.5 \tan(\alpha_1+\theta) F_1 \Big] - C(n_1)\, JJ_{21} \Big\}$$

$$g(2) = a_2 e^{ip_2} \Big\{ JJ_{12} - \Big[C(n_2) \mp B(n_2, \alpha_1+\theta) \mp 0.5 \tan(\alpha_1+\theta) F_2 \Big] - C(n_2)\, JJ_{22} \Big\}$$

$$g(3) = a_2 e^{ip_3} \Big\{ JJ_{12} - \Big[C(n_3) \mp B(n_3, \theta) \pm 0.5 \tan(\alpha_2+\theta) F_3 \Big] - C(n_3)\, JJ_{22} \Big\}$$

$$g(4) = a_4 e^{ip_4} \Big\{ JJ_{14} - \Big[C(n_4) \mp B(n_4, \alpha_2+\theta) \mp 0.5 \tan(\alpha_2+\theta) F_4 \Big] - C(n_4)\, JJ_{24} \Big\}$$

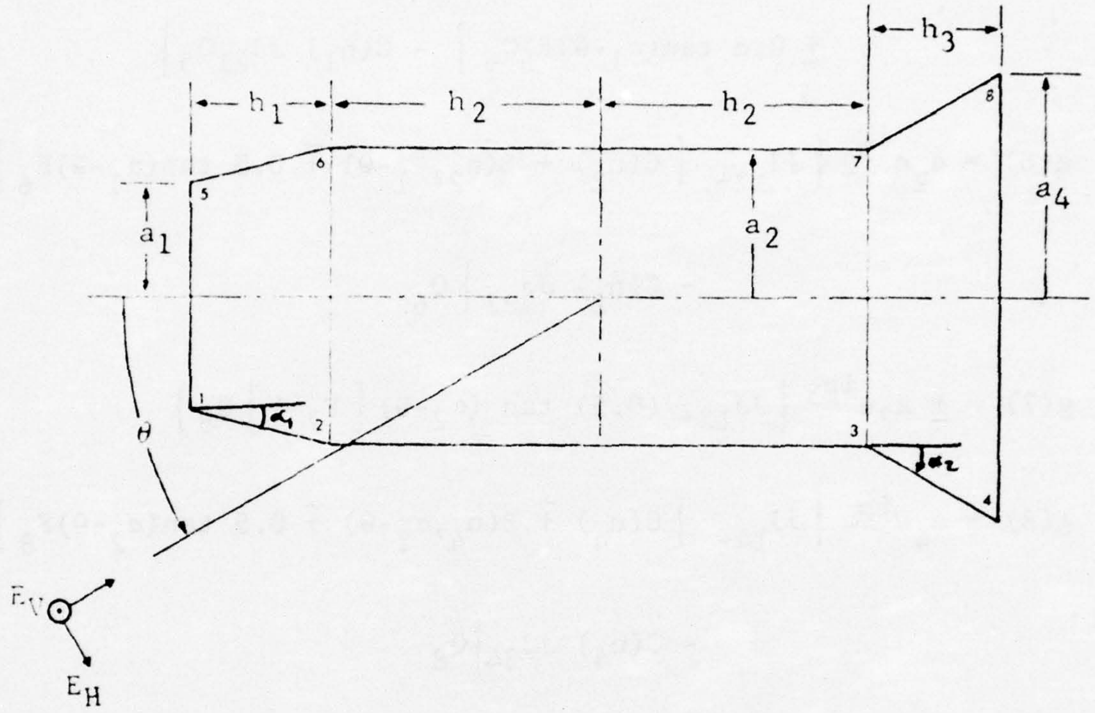


Fig. L.5-1 GEOMETRY OF FRUSTRUM-CYLINDER-FRUSTRUM

$$g(5) = a_1 e^{ip_1} \left\{ JJ_{11+} \left[C(n_1)Q_5 \mp B(n_1, \pi/2-\theta)Q_5 \pm 0.5 \tan(\alpha_1-\theta)F_5Q_6 \right] - C(n_1)\ JJ_{21}Q_5 \right\}$$

$$g(6) = a_2 e^{ip_2} \left\{ JJ_{12+} \left[C(n_2) \mp B(n_2, \alpha_1-\theta) \mp 0.5 \tan(\alpha_1-\theta)F_6 \right] - C(n_2)\ JJ_{22} \right\} Q_6$$

$$g(7) = \pm\ a_2 e^{ip_3} \left\{ JJ_{12+}\ (0.5)\ \tan\ (\alpha_2-\theta) \left[F_7-1 \right] Q_8 \right\}$$

$$g(8) = a_4 e^{ip_4} \left\{ JJ_{14+} \left[C(n_4) \mp B(n_4, \alpha_2-\theta) \mp 0.5 \tan(\alpha_2-\theta)F_8 \right] - C(n_4)\ JJ_{24} \right\} Q_8$$

For $\pi/2 \quad \theta$

$$g(1) = a_1 e^{ip_1} \left\{ JJ_{11-} \left[C(n_1) \mp B(n_1, \pi-\theta-\alpha_1) \pm 0.5 \tan(\alpha_1+\theta)F_1 \right] - C(n_1)\ JJ_{21} \right\} Q_1$$

$$g(2) = a_2 e^{ip_2} \left\{ JJ_{12-} \left[C(n_2)\ Q_2 \mp B(n_2, \pi-\theta)Q_2 \mp 0.5 \tan(\alpha_1+\theta)F_2Q_1 \right] \mp 0.5 \tan\theta(Q_2-Q_3) - C(n_2)\ JJ_{22}Q_2 \right\}$$

$$g(3) = a_2 e^{ip_3} \left\{ JJ_{12-} \left[C(n_3) \mp B(n_3, \pi - \alpha_2 - \theta) \pm 0.5 \tan\ (\alpha_2 + \theta) F_3 \right] - C(n_3) JJ_{22} \right\} Q_3$$

$$g(4) = a_2 e^{ip_4} \left\{ JJ_{14-} \left[C(n_4) \mp B(n_4, \frac{3\pi}{2} - \theta) \mp 0.5 \tan\ (\alpha_2 + \theta) F_4 Q_3 \right] - C(n_4) JJ_{24} \right\}$$

$$g(8) = a_4 e^{ip_4} \left\{ JJ_{14+} \left[C(n_4) \mp B(n_4, \theta - \pi/2) \right] - C(n_4) JJ_{24} \right\}$$

$$g(5) = g(6) = g(7) = 0$$

where the upper and lower signs in the previous expressions correspond to vertical and horizontal polarization, respectively, and

$$C(n) = \frac{\sin \frac{\pi}{n}}{n} \quad \frac{1}{\cos \pi/n - 1}$$

$$B(n, \psi) = \frac{\sin \pi/n}{n} \quad \frac{1}{\cos \pi/n - \cos \frac{2\psi}{n}}$$

$$JJ_{1m\mp} = \left[J_o(2ka_m \sin\ ) \mp i\ J_1(2ka_m \sin\ ) \right]$$

$$JJ_{2m} = \left[J_o(2ka_m \sin\ ) + J_2(2ka_m \sin\ ) \right]$$

$$n_1 = 1.5 - \frac{\alpha_1}{\pi}$$

$$n_2 = 1 + \frac{\alpha_1}{\pi}$$

$$n_3 = 1 - \frac{\alpha_2}{\pi}$$

$$n_4 = 1.5 + \frac{\alpha_2}{\pi}$$

$$p_1 = -2k(h_1 + h_2)\cos\theta$$

$$p_2 = -2k\ h_2 \cos\theta$$

$$p_3 = 2k\ h_2 \cos\theta$$

$$p_4 = 2k(h_2 + h_3)\cos\theta$$

$$Q_5 = Q(2ka_1(\pi/2-\theta))$$

$$Q_{\begin{pmatrix}6\\8\end{pmatrix}} = Q(2ka_{\begin{pmatrix}2\\4\end{pmatrix}}(\alpha_{\begin{pmatrix}1\\3\end{pmatrix}} -\theta))$$

$$Q_{\begin{pmatrix}1\\2\\3\end{pmatrix}} = Q(2ka_{\begin{pmatrix}1\\2\\2\end{pmatrix}}(\pi-\alpha_{\begin{pmatrix}1\\3\\2\end{pmatrix}} -\theta))$$

$$\tau^2{}_1 = 2ka_1 \csc{}_1 \cos(\alpha_1+\theta)$$

$$\tau^2{}_{\begin{pmatrix}3\\4\end{pmatrix}} = 2ka_{\begin{pmatrix}2\\4\end{pmatrix}} \csc{}_2 \cos(\alpha_2+\theta)$$

$$\tau^2{}_{\begin{pmatrix}5\\6\end{pmatrix}} = 2ka_{\begin{pmatrix}1\\2\end{pmatrix}} \csc{}_1 \cos(\alpha_1-\theta)$$

$$\tau^2{}_{\begin{pmatrix}7\\8\end{pmatrix}} = 2ka_{\begin{pmatrix}2\\4\end{pmatrix}} \csc{}_2 \cos(\alpha_2-\theta)$$

$$\alpha_1 = \tan^{-1}\frac{a_2-a_1}{h_1}$$

$$\alpha_2 = \tan^{-1}\frac{a_4-a_2}{h_3}$$

$$\alpha_3 = \tan^{-1} \frac{a_4 - a_2}{2h_2 + h_3}$$

$$F_m = F(\tau_m) = \frac{e^{-i\tau_m^2}}{\tau_m} \int_0^{\tau_m} e^{it^2} dt$$

$$k = 2\pi/\lambda = \text{wave number.}$$

The preceeding equation can be used in computing the first-order scattering from the target; however, it does not include the effects of multiple reflection or diffraction. The cylindrical surface between edges 2 and 3 is partially shadowed for the case of aspect angles between 150 degrees and 170 degrees, but the magnitude of this surface reflection is small; therefore, this return is formulated using the physical boundaries of the cylindrical surface and the screening functions rather than the illuminated portion of the surface. The screening functions were also used in describing the effects of shadowing upon the returns from the target edges.

L.5.1 Inputs

The subroutine inputs are read from cards or passed in a common block. The parameters passed in the common block include:

NMIN = minimum frequency number

NMAX = maximum frequency number

DF = frequency increment (in MHz)

FC = carrier frequency (in GHz)

The card inputs are the following:

	MATH SYMBOL	VARIABLE	DEFINITION	COLUMNS
Card 1	θ	Aspect	Azimuth angle (degrees)	1-7
	a_1	A1	Smallest frustrum radius (inches)	8-14

MATH SYMBOL	VARIABLE	DEFINITION	COLUMNS
a_2	A2	Radius of cylinder (inches)	15-21
a_4	A4	Largest frustrum radius (inches)	22-28
h_1	H1	Length of first frustrum (inches)	29-35
h_2	H2	Half-length of cylinder (inches)	36-42
h_3	H3	Length of second frustrum (inches)	43-49

L.5.2 Outputs

The data base output consists of four linear arrays, EVVR, EVVI, EHHR, EHHI, which contain the real and imaginary parts of the vertically and horizontally polarized backscattered fields (in meters) at frequency increments spaced DF MHz from NMIN*DF to NMAX*DF.

L.5.3 Restrictions

L.5.3.1 Physical Dimensions

All target dimensions should be large with respect to the wavelength of the illuminating field. In addition, the target dimensions should be selected such that the angles 1 and 2, of the frustra retain the basic target shape, i.e. the target does not degenerate into a cylinder. A restriction on these angles is the following:

$$15^o<\alpha_1 < 60^o$$

$$20^o<\alpha_2 < 60^o.$$

L.5.3.2 Output

The output arrays are passed in the argument list and a value is computed and stored only in locations NMIN to NMAX.

L.5.3.3 Azimuth

The azimuth angle is restricted to the region between 0 and 180 degrees. In addition the specular azimuths of 0, 90, 180, $(90 - \alpha_1)$, and $(90 - \alpha_2)$ degrees should not be used. In order to compute the response at these angles, an angular offset of approximately 0.01 degrees should be used.

L.5.4 Definitions of Selected Terms Used in Subroutines

$$\text{COFFNI} = C(n) = \frac{\sin \frac{\pi}{n}}{n} \quad \frac{1}{\cos \pi/n - 1}$$

$$\text{where } n = \quad n_1 = 1.5 - \frac{a_1}{\pi}$$

$$\text{COEF11} = \frac{1}{\cos \pi/n - \cos \frac{2\psi}{n}}$$

$$\text{where } \psi = \quad \pi/2+\theta \qquad \text{For } \theta < \pi/2$$

$$\text{COEF1} \genfrac{}{}{0pt}{}{V}{H} = C(n_1) \mp B(n_1, \pi/2+\theta) \qquad \text{For } \theta < \pi/2$$

$$\text{COEF1} \genfrac{}{}{0pt}{}{V}{H} = C(n_1) \mp B(n_1, \pi-\theta-a_1) \qquad \text{For } \theta < \pi/2$$

$$\text{PHASE } 1 = e^{ip_1}$$

$$\text{where} \quad p_1 = -2k(h_1 + h_2) \cos\theta$$

$$\text{FF} \genfrac{}{}{0pt}{}{VV}{HH} \text{ 02} = a_2 e^{ip_2} \left\{ JJ_{12-} \left[C(n_2) \mp B(n_2, a_1+\theta) \right] \right\} \text{For } \theta < \pi/2$$

$FF^{VV}_{HH}\,09 = \mp 0.5 \tan(\alpha_1+\theta)F_2\ a_2 e^{ip_2} \left\{JJ_{12-}\right\}$ For $\theta\ < \pi/2$

$FFVV17 = -\ C(n_2)\ JJ_{22}\ \ a_2 e^{ip_2}$ For $\theta\ < \pi/2$

$TERMIP = JJ_{1m\mp} = \left[J_o(2ka_m \sin\ ) \mp i\ J_1(2ka_m \sin\ )\right]$ For $\theta\ < \pi/2$

where m = 1 and the + (lower) sign is used.

L.5.5 Subroutines Used

Subfunctions:

1. Q(X) computes the exponential smoothing function
2. F(TAUS) computes the special F function

Subroutines:

BESL(SI1, XJ0, XJ1, XJ2) computes and returns

J_0(SI1) in XJ0

J_1(SI1) in XJ1

J_2(SI1) in XJ2

```
      SUBROUTINE TARGET (EVVR, EVVI, EHHR, EHHI)
C
C * * *          ST-3A -- FRUSTRA-CYLINDER-FRUSTRA (UFIMTSEV)         * * *
C
      COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, T0
C     NMIN = MINIMUM FREQUENCY SAMPLE
C     NMAX = MAXIMUM FREQUENCY SAMPLE
C     DF   = FREQUENCY INCREMENT IN MHZ
C     FC   = CARRIER FREQUENCY IN GHZ
C
      COMPLEX TERM1P, TERM1M, TERM2P, TERM2M, TERM4P, TERM4M, PHASE1,
     1 PHASE2,PHASE3, PHASE4, FFVV01, FFVV02, FFVV03, FFVV04, FFVV05,
     2 FFVV06,FFVV07, FFVV08, FFVV09, FFVV10, FFVV11, FFVV12, FFVV13,
     3 FFVV14,FFVV15, FFVV16, FFVV17, FFVV18, FFVV19, FFVV20, FFVV21,
     4 FFVV22,FFHH01, FFHH02, FFHH03, FFHH04, FFHH05, FFHH06, FFHH07,
     5 FFHH08,FFHH09, FFHH10, FFHH11, FFHH12, FFHH13, FFHH14, FFHH15,
     6 FFHH16,FFHH17, FFHH18, FFHH19, FFHH20, FFHH21, FFHH22, FFVV,
     7 FFHH,  CFVV,   CFHH,   FTAU1,  FTAU2,  FTAU3,  FTAU4,  FTAU5,
     8 FTAU6, FTAU7,  FTAU8
      COMPLEX F
C
      DIMENSION EVVR(512), EVVI(512), EHHR(512), EHHI(512)
C
C * * * ALL DIMENSIONS ARE IN INCHES AND ALL ANGLES ARE IN DEGREES   * *
C
      READ(5,1000) ASPECT,A1,A2,A4,H1,H2,H3,H4,H5,M1
 1000 FORMAT(9F7.2,I2)
      WRITE (6,1010) A1,H1,A2,H2,A4,H3
 1010 FORMAT ( 31H1 ' FRUSTRA-CYLINDER-FRUSTRA  ',/
     1           30H ' UFIMTSEV-RUCK FORMULATION ',/
     2            8H   A1 = ,F8.3, 8H   H1 = ,F8.3,/
     3            8H   A2 = ,F8.3, 8H   H2 = ,F8.3,/
     4            8H   A4 = ,F8.3, 8H   H3 = ,F8.3,/)
C
      C      = 11.80285078
      PI     = 3.14159265358979
      SPI = SQRT(PI)
      DTR    = PI / 180.0
      RTD = 180.0/PI
      WC     = 2.0 * PI * FC
      XK00 = WC/C
      X2K0 = XK00+XK00
      X2K0A1 = X2K0*A1
      X2K0A2 = X2K0*A2
      X2K0A4 = X2K0*A4
C
      THETA  = ASPECT * DTR
      STHT   = SIN(THETA)
      CTHT   = COS(THETA)
      TANATT = STHT / CTHT
C
      SHADOW = (A4 - A2) / (H2 + H2 + H3)
      SHADOW = ATAN(SHADOW)
      ALPHA1 = ATAN((A2-A1)/H1)
      ALPHA2 = ATAN((A4-A2)/H3)
      X1D = ALPHA1*RTD
      X2D = ALPHA2*RTD
      X3D = SHADOW*RTD
      WRITE (6,2010) ASPECT,X1D,X2D,X3D
 2010 FORMAT ( 18H0  ASPECT ANGLE = , F8.3,/,
     1          11H  ALPHA1 = ,F8.3,/, 11H  ALPHA2 = ,F8.3,/,
     2          19H  SHADOW(ALPHA3) = ,F8.3 )
C
      SA1 = SIN(ALPHA1)
      SA2 = SIN(ALPHA2)
      A1PT = ALPHA1+THETA
      A1MT = ALPHA1-THETA
      A2PT = ALPHA2+THETA
      A2MT = ALPHA2-THETA
      TANAP1 = TAN(A1PT)
      TANAP2 = TAN(A2PT)
      CA1PTS = 2.*COS(A1PT)/SA1
      CA2PTS = 2.*COS(A2PT)/SA2
C
      XN1    = 1.5 - (ALPHA1 / PI)
      XN2    = 1.0 + (ALPHA1 / PI)
      XN3    = 1.0 - (ALPHA2 / PI)
      XN4    = 1.5 + (ALPHA2 / PI)
      CPON1 = COS(PI/XN1)
      CPON2 = COS(PI/XN2)
      CPON3 = COS(PI/XN3)
      CPON4 = COS(PI/XN4)
      TERM01 = (SIN(PI / XN1)) / XN1
      TERM02 = (SIN(PI / XN2)) / XN2
      TERM03 = (SIN(PI / XN3)) / XN3
      TERM04 = (SIN(PI / XN4)) / XN4
C     COEFNX ARE C(NX) TERMS
      COEFN1= TERM01/(CPON1 - 1.)
      COEFN2= TERM02/(CPON2 - 1.)
      COEFN3= TERM03/(CPON3 - 1.)
      COEFN4= TERM04/(CPON4 - 1.)
C
      IF (ASPECT .GT. 90.0) GO TO 10
C
C     DIFFRACTION TERMS ( C(N)-/+B(N,PHI) = COEFXX TERMS
C          COMPUTED HERE FOR THETA LT. 90
      COEF11 = 1.0 / ( CPON1 - COS((THETA+THETA+PI)/XN1))
      COEF12 = 1.0 / ( CPON2 - COS((A1PT + A1PT)   /XN2))
      COEF13 = 1.0 / ( CPON3 - COS((THETA + THETA) /XN3))
      COEF14 = 1.0 / ( CPON4 - COS((A2PT + A2PT)   /XN4))
      COEF21 = 1.0 / ( CPON1 - COS((THETA+THETA-PI)/XN1))
      COEF22 = 1.0 / ( CPON2 - COS((A1MT + A1MT)   /XN2))
      COEF24 = 1.0 / ( CPON4 - COS((A2MT + A2MT)   /XN4))
C
      COEF1V = COEFN1 - COEF11 * TERM01
      COEF1H = COEFN1 + COEF11 * TERM01
      COEF2V = COEFN2 - COEF12 * TERM02
      COEF2H = COEFN2 + COEF12 * TERM02
      COEF3V = COEFN3 - COEF13 * TERM03
      COEF3H = COEFN3 + COEF13 * TERM03
      COEF4V = COEFN4 - COEF14 * TERM04
      COEF4H = COEFN4 + COEF14 * TERM04
      COEF5V = COEFN1 - COEF21 * TERM01
      COEF5H = COEFN1 + COEF21 * TERM01
```

L-80

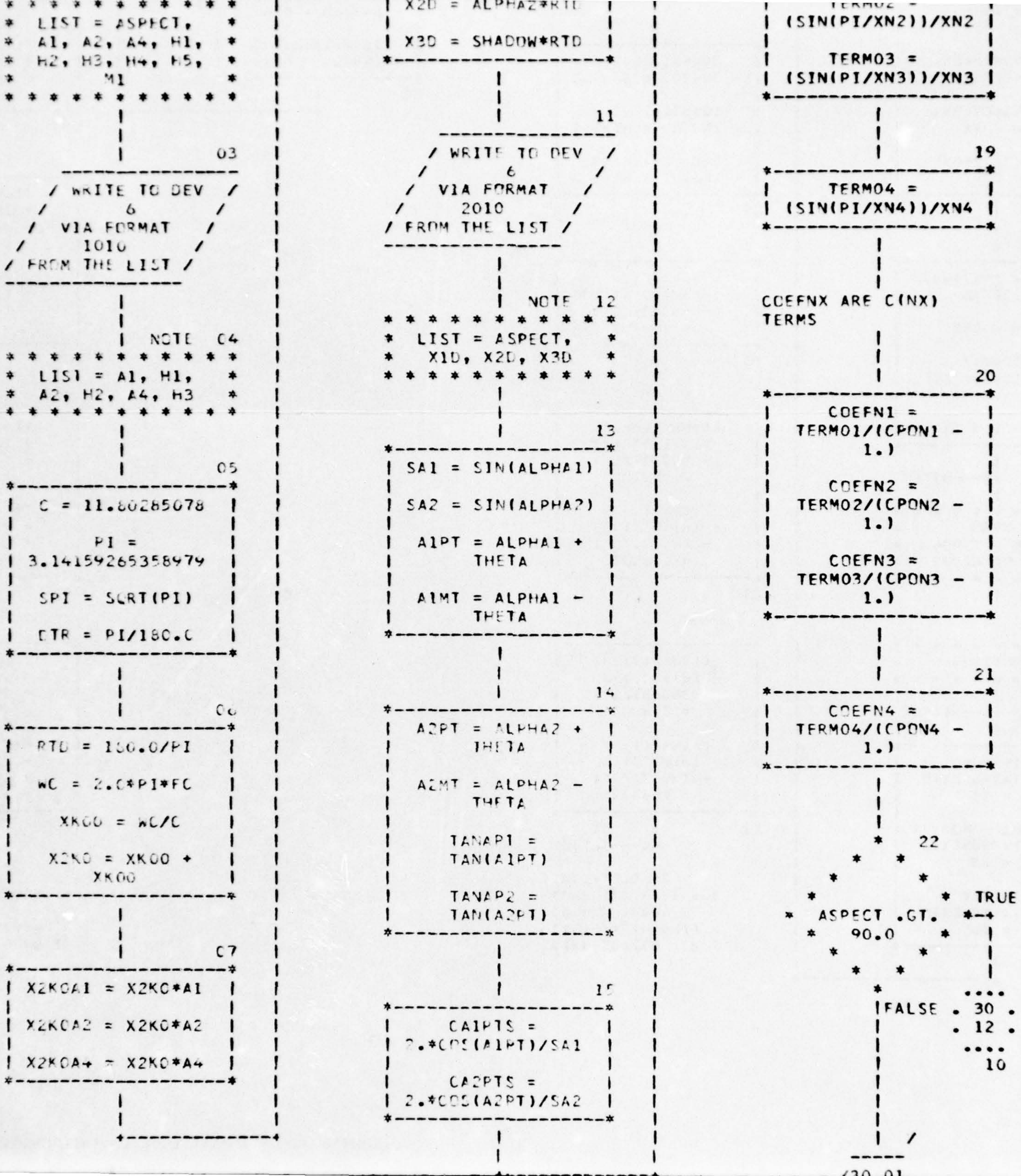
```
      COEF6V = COEFN2 - COEF22 * TERM02
      COEF6H = COEFN2 + COEF22 * TERM02
      COEF7V = COEFN4 - COEF24 * TERM04
      COEF7H = COEFN4 + COEF24 * TERM04
C
      TANAM1 = TAN(A1MT)
      TANAM2 = TAN(A2MT)
      CA1MTS = 2.*COS(A1MT)/SA1
      CA2MTS = 2.*COS(A2MT)/SA2
      Q6=Q(X2K0A2 * A1MT                 )
      Q8=Q(X2K0A4 * (SHADOW - THETA)     )
      GO TO 20
C
C     DIFFRACTION TERMS ( C(N)-/+B(N,PHI) = COEFXX TERMS
C          COMPUTED HERE FOR THETA GT. 90
   10 COEF31   = 1.0 / (CPON1 - COS(2.0*(PI-A1PT)/XN1))
      COEF32   = 1.0 / (CPON2 - COS(2.0*(PI-THETA)/XN2))
      COEF33   = 1.0 / (CPON3 - COS(2.0*(PI-A2PT)/XN3))
      COEF34   = 1.0 / (CPON4 - COS((2.0*THETA - 3.0*PI)/XN4))
      COEF44   = 1.0 / (CPON4 - COS((2.0*THETA -    PI   )/XN4))
C
      COEF1V = COEFN1 - COEF31 * TERM01
      COEF1H = COEFN1 + COEF31 * TERM01
      COEF2V = COEFN2 - COEF32 * TERM02
      COEF2H = COEFN2 + COEF32 * TERM02
      COEF3V = COEFN3 - COEF33 * TERM03
      COEF3H = COEFN3 + COEF33 * TERM03
      COEF4V = COEFN4 - COEF34 * TERM04
      COEF4H = COEFN4 + COEF34 * TERM04
      COEF5V = COEFN4 - COEF44 * TERM04
      COEF5H = COEFN4 + COEF44 * TERM04
C
      Q1=Q(X2K0A1 * (PI-A1PT)            )
      Q2=Q(X2K0A2 * (PI - SHADOW - THETA))
      Q3=Q(X2K0A2 * (PI-A2PT)            )
   20 CONTINUE
C
      DO 900 I = NMIN, NMAX
      XI     = I - 1
      W      = (2. * PI * XI * DF) / 1000.0
      XK0    = W / C
      XK02C    = XK0 * (CTHT + CTHT)
C
      TAU1     = XK0 * A1 * CA1PTS
      TAU2     = XK0 * A2 * CA1PTS
      TAU3     = XK0 * A2 * CA2PTS
      TAU4     =XK0 * A4 * CA2PTS
      FTAU1  = F     (TAU1)
      FTAU2  = F     (TAU2)
      FTAU3  = F     (TAU3)
      FTAU4  = F     (TAU4)
C
      SI1    = 2.0 * XK0 * A1 * STHT
      SI2    = 2.0 * XK0 * A2 * STHT
      SI4    = 2.0 * XK0 * A4 * STHT
      CALL BESL  (SI1, XJ0X1, XJ1X1, XJ2X1)
      CALL BESL  (SI2, XJ0X2, XJ1X2, XJ2X2)
      CALL BESL  (SI4, XJ0X4, XJ1X4, XJ2X4)
      TERM1P = CMPLX(XJ0X1, XJ1X1)
      TERM1M = CONJG(TERM1P)
      TERM2P = CMPLX(XJ0X2, XJ1X2)
      TERM2M = CONJG(TERM2P)
      TERM4P = CMPLX(XJ0X4, XJ1X4)
      TERM4M = CONJG(TERM4P)
      TERM5  = XJ0X1 + XJ2X1
      TERM6  = XJ0X2 + XJ2X2
      TERM7  = XJ0X4 + XJ2X4
C
      PHI1      = XK02C * (H1+H2)
      PHI2      = XK02C * (H2)
      PHI3      = PHI2
      PHI4      = XK02C * (H2+H3)
      PHASE1 = CMPLX(COS(PHI1), -SIN(PHI1))
      PHASE2 = CMPLX(COS(PHI2), -SIN(PHI2))
      PHASE3 = CMPLX(COS(PHI3),  SIN(PHI3))
      PHASE4 = CMPLX(COS(PHI4),  SIN(PHI4))
C
      IF (ASPECT .GT. 90.0) GO TO 30
C
      TAU5      = XK0 * A1 * CA1MTS
      TAU6      = XK0 * A2 * CA1MTS
      TAU7      = XK0 * A2 * CA2MTS
      TAU8      = XK0 * A4 * CA2MTS
      FTAU5  = F      (TAU5)
      FTAU6  = F      (TAU6)
      FTAU7  = F      (TAU7)
      FTAU8  = F      (TAU8)
C
      FFVV01 = A1 * TERM1M * COEF1V * PHASE1
      FFHH01 = A1 * TERM1M * COEF1H * PHASE1
      FFVV02 = A2 * TERM2M * COEF2V * PHASE2
      FFHH02 = A2 * TERM2M * COEF2H * PHASE2
      FFVV03 = A2 * TERM2M * COEF3V * PHASE3
      FFHH03 = A2 * TERM2M * COEF3H * PHASE3
      FFVV04 = A4 * TERM4M * COEF4V * PHASE4
      FFHH04 = A4 * TERM4M * COEF4H * PHASE4
      FFVV05 = A1 * TERM1P * COEF5V * PHASE1
      FFHH05 = A1 * TERM1P * COEF5H * PHASE1
      FFVV06 = A2 * TERM2P * COEF6V * PHASE2 * Q6
      FFHH06 = A2 * TERM2P * COEF6H * PHASE2 * Q6
      FFVV07 = A4 * TERM4P * COEF7V * PHASE4 * Q8
      FFHH07 = A4 * TERM4P * COEF7H * PHASE4 * Q8
      FFVV08 = A1 * TERM1M * TANAP1 * ( 0.5) * FTAU1 * PHASE1
      FFHH08 = -FFVV08
      FFVV09 = A2 * TERM2M * TANAP1 * (-0.5) * FTAU2 * PHASE2
      FFHH09 = -FFVV09
      FFVV10 = A2 * TERM2M * TANAP2 * ( 0.5) * FTAU3 * PHASE3
      FFHH10 = -FFVV10
      FFVV11 = A4 * TERM4M * TANAP2 * (-0.5) * FTAU4 * PHASE4
      FFHH11 = -FFVV11
      FFVV12 = A1 * TERM1P * TANAM1 * ( 0.5) * FTAU5 * PHASE1 * Q6
      FFHH12 = -FFVV12
      FFVV13 = A2 * TERM2P * TANAM1 * (-0.5) * FTAU6 * PHASE2 * Q6
      FFHH13 = -FFVV13
      FFVV14 = A2 * TERM2P * TANAM2 *(-.5)*(1.-FTAU7)* PHASE3 * Q8
      FFHH14 = -FFVV14
      FFVV15 = A4 * TERM4P * TANAM2 * (-0.5) * FTAU8 * PHASE4 * Q8
      FFHH15 = -FFVV15
      FFVV16 = -A1 * COEFN1 * TERM5  * PHASE1
      FFHH16 =  FFVV16
      FFVV17 = -A2 * COEFN2 * TERM6  * PHASE2
      FFHH17 =  FFVV17
      FFVV18 = -A2 * COEFN3 * TERM6  * PHASE3
      FFHH18 =  FFVV18
      FFVV19 = -A4 * COEFN4 * TERM7  * PHASE4
      FFHH19 =  FFVV19
      FFVV20 = -A1 * COEFN1 * TERM5  * PHASE1
      FFHH20 =  FFVV20
      FFVV21 = -A2 * COEFN2 * TERM6  * PHASE2 * Q6
      FFHH21 =  FFVV21
      FFVV22 = -A4 * COEFN4 * TERM7  * PHASE4 * Q8
      FFHH22 =  FFVV22
C
      FFVV   = FFVV01 + FFVV02 + FFVV03 + FFVV04 + FFVV05 + FFVV06 +
     1         FFVV07 + FFVV08 + FFVV09 + FFVV10 + FFVV11 + FFVV12 +
     2         FFVV13 + FFVV14 + FFVV15 + FFVV16 + FFVV17 + FFVV18 +
     3         FFVV19 + FFVV20 + FFVV21 + FFVV22
      FFHH   = FFHH01 + FFHH02 + FFHH03 + FFHH04 + FFHH05 + FFHH06 +
     1         FFHH07 + FFHH08 + FFHH09 + FFHH10 + FFHH11 + FFHH12 +
     2         FFHH13 + FFHH14 + FFHH15 + FFHH16 + FFHH17 + FFHH18 +
     3         FFHH19 + FFHH20 + FFHH21 + FFHH22
C
      GO TO 40
C
   30 CONTINUE
      FFVV01 = A1 * TERM1M * COEF1V * PHASE1 * Q1
      FFHH01 = A1 * TERM1M * COEF1H * PHASE1 * Q1
      FFVV02 = A2 * TERM2M * COEF2V * PHASE2 * Q2
      FFHH02 = A2 * TERM2M * COEF2H * PHASE2 * Q2
      FFVV03 = A2 * TERM2M * COEF3V * PHASE3 * Q3
      FFHH03 = A2 * TERM2M * COEF3H * PHASE3 * Q3
      FFVV04 = A4 * TERM4M * COEF4V * PHASE4
      FFHH04 = A4 * TERM4M * COEF4H * PHASE4
      FFVV05 = A4 * TERM4P * COEF5V * PHASE4
      FFHH05 = A4 * TERM4P * COEF5H * PHASE4
      FFVV06 = A1 * TERM1M * TANAP1 * ( 0.5) * FTAU1 * PHASE1 * Q1
      FFHH06 = -FFVV06
      FFVV07 = A2 * TERM2M * TANAP1 * (-0.5) * FTAU2 * PHASE2 * Q1
      FFHH07 = -FFVV07
      FFVV08 = A2 * TERM2M * TANAP2 * ( 0.5) * FTAU3 * PHASE3 * Q3
      FFHH08 = -FFVV08
      FFVV09 = A2 * TERM2M * TANATT * ( 0.5)         * PHASE3 *(Q2-Q3)
      FFHH09 = -FFVV09
      FFVV11 = A4 * TERM4M * TANAP2 * (-0.5) * FTAU4 * PHASE4 * Q3
      FFHH11 = -FFVV11
      FFVV12 = -A1 * COEFN1 * TERM5  * PHASE1 * Q1
      FFHH12 =  FFVV12
      FFVV13 = -A2 * COEFN2 * TERM6  * PHASE2 * Q2
      FFHH13 =  FFVV13
      FFVV14 = -A2 * COEFN3 * TERM6  * PHASE3 * Q3
      FFHH14 =  FFVV14
      FFVV15 = -A4 * COEFN4 * TERM7  * PHASE4 *Q8
      FFHH15 =  FFVV15
      FFVV16 = -A4 * COEFN4 * TERM7  * PHASE4
      FFHH16 =  FFVV16
C
      FFVV   = FFVV01 + FFVV02 + FFVV03 + FFVV04 + FFVV05 + FFVV06 +
     1         FFVV07 + FFVV08 + FFVV09          + FFVV11 + FFVV12 +
     2         FFVV13 + FFVV14 + FFVV15 + FFVV16
C
      FFHH   = FFHH01 + FFHH02 + FFHH03 + FFHH04 + FFHH05 + FFHH06 +
     1         FFHH07 + FFHH08 + FFHH09          + FFHH11 + FFHH12 +
     2         FFHH13 + FFHH14 + FFHH15 + FFHH16
C
C
   40 FFVV   = FFVV * .02539998 * SPI
      FFHH   = FFHH * .02539998 * SPI
C
      CFVV   =-CONJG(FFVV)
      CFHH   = CONJG(FFHH)
C
      EVVR(I) =REAL(CFVV)
      EVVI(I) =AIMAG(CFVV)
      EHHR(I) =REAL(CFHH)
      EHHI(I) =AIMAG(CFHH)
C
  900 CONTINUE
C
      RETURN
      END
      SUBROUTINE BESL ( X, B0, B1, B2 )
C
C   * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS
C   * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS
C   * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )
C
      S = 1.0
      IF (X .LT. 0.0) S=-1.0
      X = ABS(X)
C
      IF ( X .GT. 1.E-6 ) GO TO 5
      B0 = 1.0
      B1 = 0.0
      B2 = 0.0
      X = X * S
      RETURN
C
    5 CONTINUE
C
    1 IF ( X .GE. 3.) GO TO 9
      X1 = X/3.
      X1 = X1*X1
      B = 1. + X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479
     1 +       X1*(-.0039444+ X1*2.1E-4 )))) )
      GO TO 10
C
    9 X2 = 3./X
      F0 =   .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*
     1      (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )
      T0 = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573
     1     +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))
      B = F0*COS(T0)/SQRT(X)
```

```
10 B0 = B

 2 IF ( X .GE. 3. ) GO TO 19
   X1 = X/3.
   X1 = X1*X1
   B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*
  1               (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4)))))  )
   GO TO 20

19 X2 = 3./X
   F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*
  1       (-.00249511 +X2*(.00113653 -.00020033*X2 )))))
   T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879
  1     +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))
   B = F1*COS(T1)/SQRT(X)

20 B1 = B * S
   X = X * S
   B2= (2./X)*B1 - B0
50 RETURN
   END
   COMPLEX FUNCTION F(TAU)

   COMPUTES FTAU WHERE FTAU =(EXP(-J*TAU**2)/2*TAU)*SQRT(PI/2.)*
                          (C2(TAU**2)  + J*S2(TAU**2))

   COMPLEX B, FP
   PI = 3.14159265358979
   PIO2 = PI/2.
   C1 = SQRT(PI/2.)
   C2 = 1./C1
   ATAUS = ABS(TAU)
   IF (ATAUS .LE. 0.5 )GO TO 20

   FOR TAUS .GT. 0.5, FUNCTION COMPUTED USING POLYNOMIAL
                       APPROXIMATION
 + REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,
 +            SECTIONS  7.3.  , 7.3.10, 7.3.32,7.3.33)
   TAUS = SQRT(ATAUS)
   X = C2*TAUS
   XS = X*X

   FX = (1.0+0.926*X)/(2.0+1.792*X+3.104*XS)
   GX = 1.0/(2.0+4.142*X+3.492*XS+6.67*X*XS)

   CC1XS = COS(ATAUS)
   SC1XS = SIN(ATAUS)

   CX = 0.5 + FX*SC1XS - GX*CC1XS
   SX = 0.5 - FX*CC1XS - GX*SC1XS

   IF (TAU .LT. 0.0) GO TO 10
   B = CMPLX(CX,SX)
   FP = CMPLX(  COS(ATAUS), -SIN(ATAUS) )
   F = (C1*B*FP)/TAUS
   RETURN
   10 CONTINUE
      B = CMPLX(SX,CX)
      A = ATAUS-PIO2
      FP = CMPLX( COS(A),SIN(A) )
      F = (B*FP*C1)/TAUS
      RETURN
C
   20 CONTINUE
C     FOR TAUS .LE. 0.5, FUNCTION IS EXPANDED IN SERIES AND FIRST FEW
C                         TERMS INTEGRATED TERM BY TERM TO OBTAIN RESULT
      FP = CMPLX(COS(TAU),-SIN(TAU))
      TS = TAU*TAU
      FR = 1 - TS*( .1 - .0046296296*TS)
      FI = TAU *( .333333333 - TS*(.0238095238 - 7.57575757E-4*TS))
      B = CMPLX(FR,FI)
      F = FP*B
      RETURN
      END
      FUNCTION Q(Z)
C     Q(Z) = 0.5*(1 + ERF(Z))
C   * ERF(Z) IS EVALUATED USING A RATIONAL POLYNOMIAL APPROXIMATION
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,
C   *            SECTION 7.1.26)
C
      IF ( Z.GT. 2.) GO TO 10
      IF ( Z.LT.-2.) GO TO 20
      AZ = ABS(Z)
      P = 1.0/(1.0 + .47047*AZ)
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)
      IF (Z) 2,4,6
    2 Q = (1.0 - Y)/2.
      RETURN
    4 Q = .5
      RETURN
    6 Q = (1.0 + Y)/2.
      RETURN
   10 Q = 1.
      RETURN
   20 Q = 0.
      RETURN
      END
```

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI)

```
    -----------
   / TARGET  /
  -----------

       *                 *------------------>*          *------------------>*
                         |                   |          |                   |
                         |                   |      08  |                   |      16
  * * *         ST-3A    |  *-------------------*       |  *-------------------*
  --                     |  |    THETA =        |       |  |   XN1 = 1.5 -     |
  FRUSTRA-CYLINDER-      |  |    ASPECT*DTR     |       |  |   (ALPHA1/PI)     |
  FRUSTRA (UFIMTSEV)     |  |                   |       |  |                   |
   * * *                 |  | STHT = SIN(THETA) |       |  |   XN2 = 1.0 +     |
                         |  |                   |       |  |   (ALPHA1/PI)     |
                         |  | CTHT = COS(THETA) |       |  |                   |
                         |  |                   |       |  |   XN3 = 1.0 -     |
  NMIN = MINIMUM         |  |    TANATT =       |       |  |   (ALPHA2/PI)     |
  FREQUENCY SAMPLE       |  |    STHT/CTHT      |       |  |                   |
  NMAX = MAXIMUM         |  *-------------------*       |  |   XN4 = 1.5 +     |
  FREQUENCY SAMPLE       |            |                 |  |   (ALPHA2/PI)     |
  DF    = FREQUENCY      |            |                 |  *-------------------*
  INCREMENT IN MHZ       |            |          09     |            |
  FC    = CARRIER        |  *-------------------*       |            |
  FREQUENCY IN GHZ       |  |  SHADOW = (A4 -   |       |            |          17
                         |  |  A2)/(H2 + H2 +   |       |  *-------------------*
                         |  |       H3)         |       |  |     CPON1 =       |
                         |  |                   |       |  |   COS(PI/XN1)     |
  * * * ALL DIMENSIONS   |  |    SHADOW =       |       |  |                   |
  ARE IN INCHES AND ALL  |  |  ATAN(SHADOW)     |       |  |     CPON2 =       |
  ANGLES ARE IN DEGREES  |  |                   |       |  |   COS(PI/XN2)     |
   * *                   |  |    ALPHA1 =       |       |  |                   |
                         |  |   ATAN((A2 -      |       |  |     CPON3 =       |
        *                |  |     A1)/H1)       |       |  |   COS(PI/XN3)     |
        |         01     |  *-------------------*       |  |                   |
   ----------------      |            |                 |  |     CPON4 =       |
   / READ FROM DEV /     |            |                 |  |   COS(PI/XN4)     |
  /      5        /      |            |          10     |  *-------------------*
 /  VIA FORMAT   /       |  *-------------------*       |            |
/     1000      /        |  |    ALPHA2 =       |       |            |
/ INTO THE LIST /        |  |   ATAN((A4 -      |       |            |          18
----------------         |  |     A2)/H3)       |       |  *-------------------*
        |                |  |                   |       |  |     TERM01 =      |
        |                |  | X1D = ALPHA1*RTD  |       |  | (SIN(PI/XN1))/XN1 |
        |   NOTE  02     |  |                   |       |  |                   |
* * * * * * * * * * * *  |  | X2D = ALPHA2*RTD  |       |  |     TERM02 =      |
*  LIST = ASPECT,      * |  |                   |       |  | (SIN(PI/XN2))/XN2 |
```

2



```
* * * * * * * * * * * *
*  LIST = ASPECT,     *
*  A1, A2, A4, H1,    *
*  H2, H3, H4, H5,    *
*        M1           *
* * * * * * * * * * * *

                     03
   / WRITE TO DEV  /
   /      6        /
   /  VIA FORMAT   /
   /     1010      /
   / FROM THE LIST /

              NOTE  04
* * * * * * * * * * * *
*  LIST = A1, H1,     *
*  A2, H2, A4, H3     *
* * * * * * * * * * * *

                     05
| C = 11.80285078     |
|       PI =          |
| 3.14159265358979    |
|  SPI = SQRT(PI)     |
|  DTR = PI/180.0     |

                     06
| RTD = 180.0/PI      |
| WC = 2.0*PI*FC      |
|   XK00 = WC/C       |
|  X2K0 = XK00 +      |
|      XK00           |

                     07
| X2K0A1 = X2K0*A1    |
| X2K0A2 = X2K0*A2    |
| X2K0A4 = X2K0*A4    |


| X2D = ALPHA2*RTD    |
| X3D = SHADOW*RTD    |

                     11
   / WRITE TO DEV  /
   /      6        /
   /  VIA FORMAT   /
   /     2010      /
   / FROM THE LIST /

              NOTE  12
* * * * * * * * * * * *
*  LIST = ASPECT,     *
*   X1D, X2D, X3D     *
* * * * * * * * * * * *

                     13
| SA1 = SIN(ALPHA1)   |
| SA2 = SIN(ALPHA2)   |
|  A1PT = ALPHA1 +    |
|      THETA          |
|  A1MT = ALPHA1 -    |
|      THETA          |

                     14
|  A2PT = ALPHA2 +    |
|      THETA          |
|  A2MT = ALPHA2 -    |
|      THETA          |
|    TANAP1 =         |
|    TAN(A1PT)        |
|    TANAP2 =         |
|    TAN(A2PT)        |

                     15
|     CA1PTS =        |
| 2.*COS(A1PT)/SA1    |
|     CA2PTS =        |
| 2.*COS(A2PT)/SA2    |


|      TERMO2 =       |
| (SIN(PI/XN2))/XN2   |
|      TERMO3 =       |
| (SIN(PI/XN3))/XN3   |

                     19
|      TERMO4 =       |
| (SIN(PI/XN4))/XN4   |

COEFNX ARE C(NX)
TERMS

                     20
|      COEFN1 =       |
|  TERMO1/(CPON1 -    |
|      1.)            |
|      COEFN2 =       |
|  TERMO2/(CPON2 -    |
|      1.)            |
|      COEFN3 =       |
|  TERMO3/(CPON3 -    |
|      1.)            |

                     21
|      COEFN4 =       |
|  TERMO4/(CPON4 -    |
|      1.)            |

            *    22
         *     *
      *           *  TRUE
   *  ASPECT .GT.  *-+
      *   90.0    *  |
         *     *     |
            *      ....
            |FALSE . 30 .
            |      . 12 .
            |       ....
            |        10
            |  /
           ----
          (30 01
```

L-87

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI)

L-88

```
 29.22--->*                                                                           +---------------------->*
            |                                                                         |                       |         19
DIFFRACTION TERMS (                                          DIFFRACTION TERMS (      |          *-----------------------*
C(N)-/+B(N,PHI) =                                            C(N)-/+B(N,PHI) =        |          |         Q1 =          |
COEFXX TERMS                                                 COEFXX TERMS             |          |  Q(X2KOA1*(PI -       |
COMPUTED HERE FOR                                            COMPUTED HERE FOR        |          |       A1PT))          |
THETA.LT. 90                                                 THETA.GT. 90             |          |                       |
            |                                                                         |          |        Q2 =           |
            |                                                 29.22--->*              |          |  Q(X2KOA2*(PI -       |
            |         01                                           10    |          12 |          | SHADOW - THETA))      |
*----------------------*                                     *----------------------* |          |                       |
|      COEF11 =        |                                     |      COEF31 =        | |          |        Q3 =           |
|   1.0/(CPON1 -       |                                     |   1.0/(CPON1 -       | |          |  Q(X2KOA2*(PI -       |
|   COS((THETA +       |                                     |   COS(2.0*(PI -      | |          |       A2PT))          |
| THETA + PI)/XN1))    |                                     |    A1PT)/XN1))       | |          *-----------------------*
|                      |                                     |                      | |                      |
|      COEF12 =        |                                     |      COEF32 =        | |          30.11--->|
|   1.0/(CPON2 -       |                                     |   1.0/(CPON2 -       | |            20       |  NOTE  20
|    COS((A1PT +       |                                     |   COS(2.0*(PI -      | |          * * * * * * * * * * * * *
|     A1PT)/XN2))      |                                     |    THETA)/XN2))      | |          *       CONTINUE        *
*----------------------*                                     *----------------------* |          * * * * * * * * * * * * *
            |                                                            |            |                      |
            |                                                            |            |                      |
            |         02                                                 |          13 |                      |  NOTE  21
*----------------------*                                     *----------------------* |          * * * * * * * * * * * * *
|      COEF13 =        |                                     |      COEF33 =        | |          *    BEGIN DO LOOP      *
|   1.0/(CPON3 -       |                                     |   1.0/(CPON3 -       | |          *    900 I = NMIN,      *
|   COS((THETA +       |                                     |   COS(2.0*(PI -      | |          *         NMAX          *
|    THETA)/XN3))      |                                     |    A2PT)/XN3))       | |          * * * * * * * * * * * * *
|                      |                                     |                      | |                      |
|      COEF14 =        |                                     |      COEF34 =        | |          33.10--->|
|   1.0/(CPON4 -       |                                     |   1.0/(CPON4 -       | |                      |           22
|    COS((A2PT +       |                                     | COS((2.0*THETA -     | |          *-----------------------*
|     A2PT)/XN4))      |                                     |    3.0*PI)/XN4))     | |          |      XI = I - 1       |
*----------------------*                                     *----------------------* |          |                       |
            |                                                            |            |          |         W =           |
            |                                                            |            |          |    (2.*PI*XI*DF)      |
            |         03                                                 |          14 |          |       /1000.0         |
*----------------------*                                     *----------------------* |          |                       |
|      COEF21 =        |                                     |      COEF44 =        | |          |      XKO = W/C        |
|   1.0/(CPON1 -       |                                     |   1.0/(CPON4 -       | |          |                       |
|   COS((THETA +       |                                     | COS((2.0*THETA -     | |          |       XKO2C =         |
| THETA - PI)/XN1))    |                                     |      PI)/XN4))       | |          | XKO*(CTHT + CTHT)     |
|                      |                                     *----------------------* |          *-----------------------*
|      COEF22 =        |                                                 |            |                      |
|   1.0/(CPON2 -       |                                                 |            |                      |
|    COS((A1MT +       |                                                 |          15 |                      |           23
|     A1MT)/XN2))      |                                     *----------------------* |          *-----------------------*
*----------------------*                                     | COEF1V = COEFN1 -    | |          |        TAU1 =         |
            |                                                |   COEF31*TERM01      | |          |   XKO*A1*CA1PTS       |
            |                                                |                      | |          |                       |
            |         04                                     | COEF1H = COEFN1 +    | |          |        TAU2 =         |
*----------------------*    +---------------------->*        |   COEF31*TERM01      | |          |   XKO*A2*CA1PTS       |
```

2

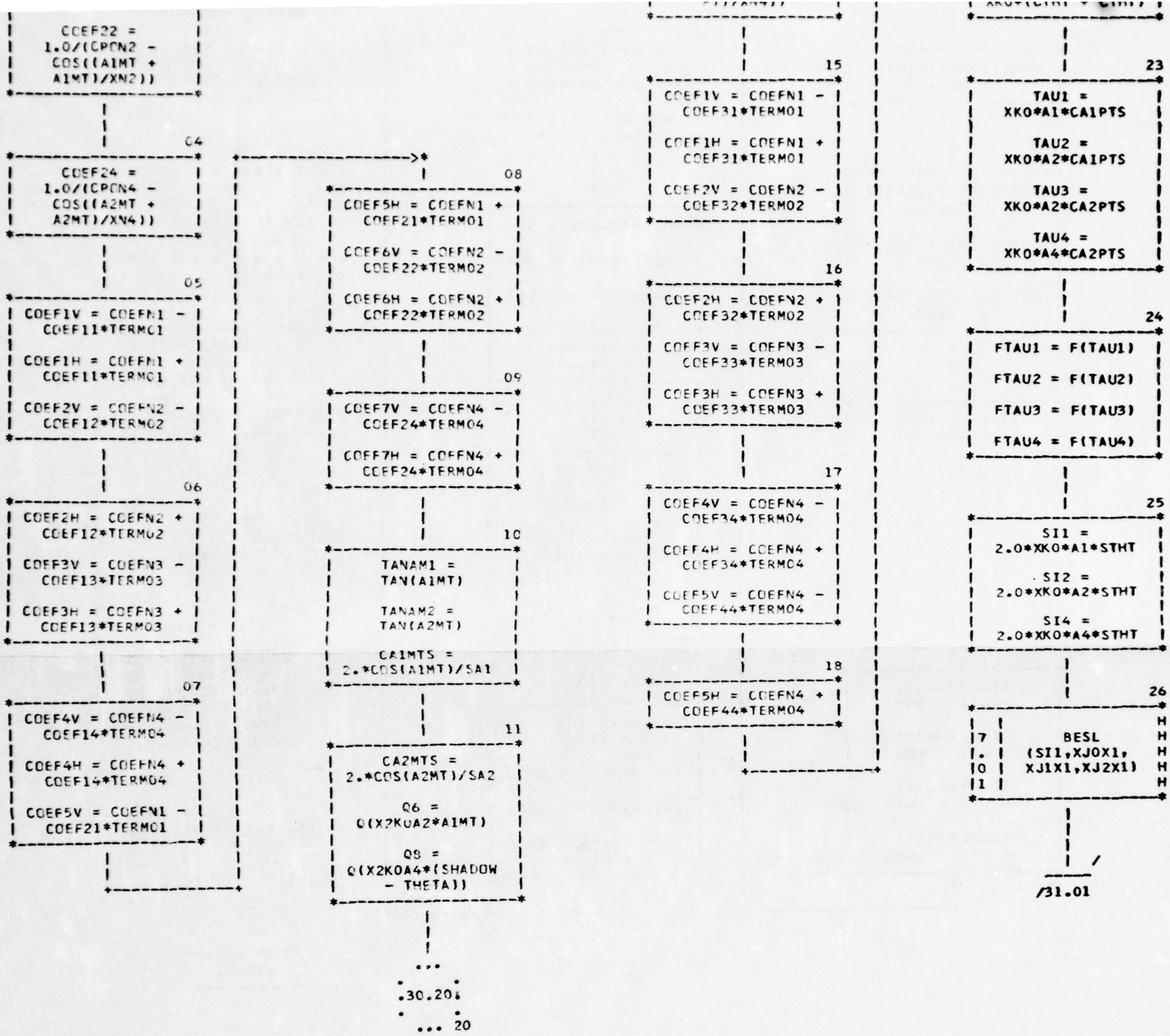
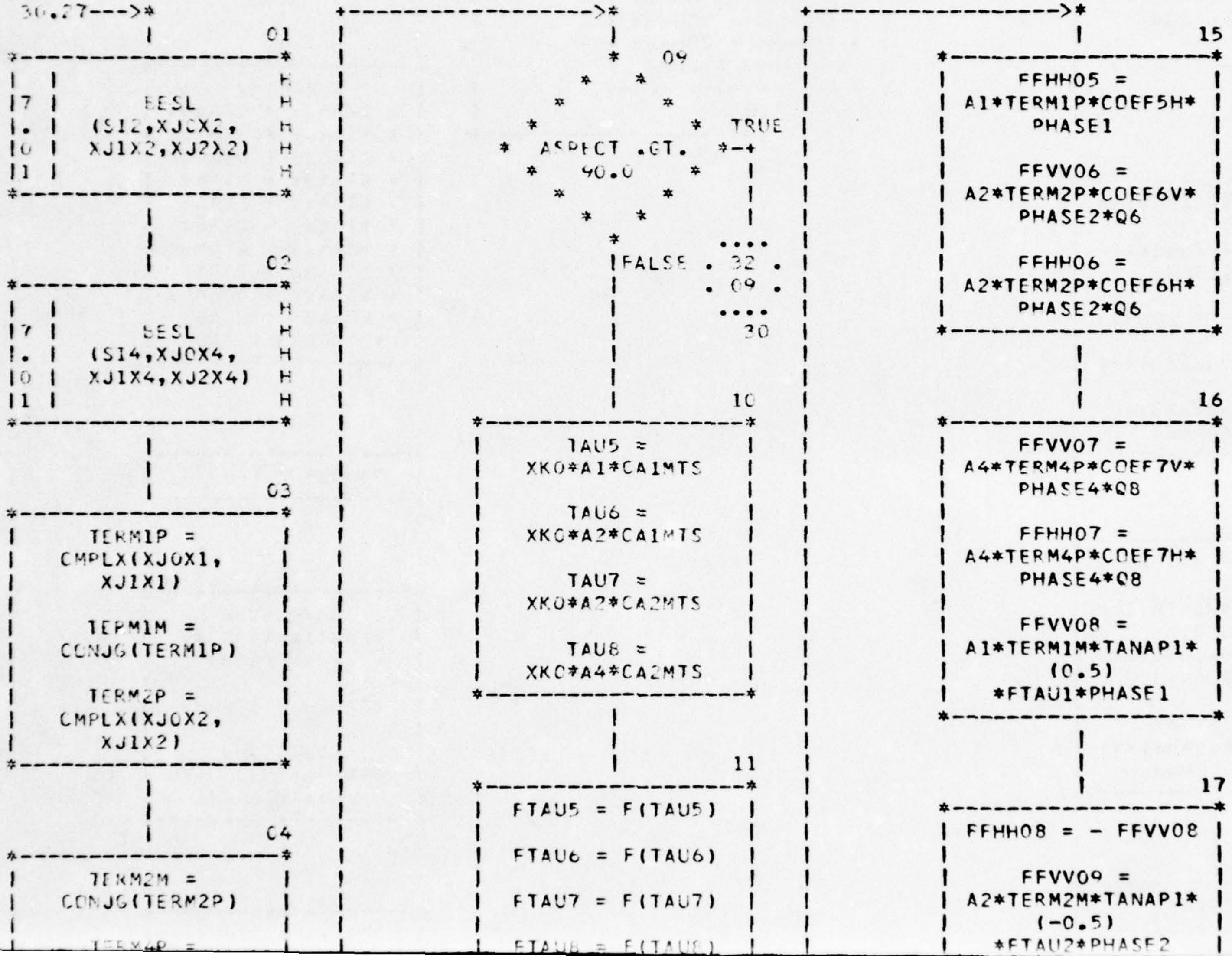
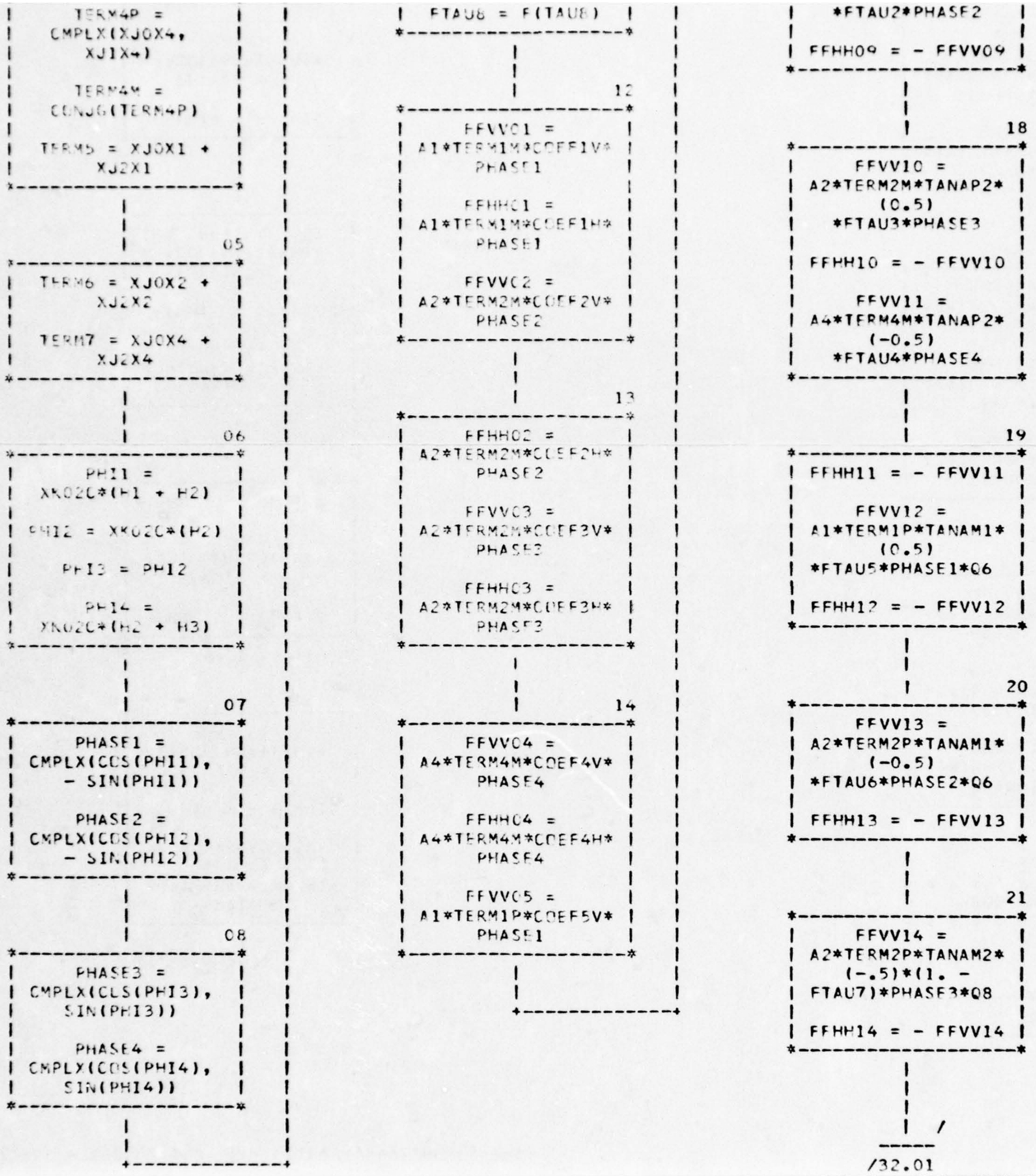

COEF22 =
1.0/(CPON2 -
COS((A1MT +
A1MT)/XN2))
04
COEF24 =
1.0/(CPON4 -
COS((A2MT +
A2MT)/XN4))
05
COEF1V = COEFN1 -
COEF11*TERMO1
COEF1H = COEFN1 +
COEF11*TERMO1
COEF2V = COEFN2 -
COEF12*TERMO2
06
COEF2H = COEFN2 +
COEF12*TERMO2
COEF3V = COEFN3 -
COEF13*TERMO3
COEF3H = COEFN3 +
COEF13*TERMO3
07
COEF4V = COEFN4 -
COEF14*TERMO4
COEF4H = COEFN4 +
COEF14*TERMO4
COEF5V = COEFN1 -
COEF21*TERMO1
08
COEF5H = COEFN1 +
COEF21*TERMO1
COEF6V = COEFN2 -
COEF22*TERMO2
COEF6H = COEFN2 +
COEF22*TERMO2
09
COEF7V = COEFN4 -
COEF24*TERMO4
COEF7H = COEFN4 +
COEF24*TERMO4
10
TANAM1 =
TAN(A1MT)
TANAM2 =
TAN(A2MT)
CA1MTS =
2.*COS(A1MT)/SA1
11
CA2MTS =
2.*COS(A2MT)/SA2
Q6 =
Q(X2KOA2*A1MT)
Q8 =
Q(X2KOA4*(SHADOW
- THETA))
.30.20.
... 20
15
COEF1V = COEFN1 -
COEF31*TERMO1
COEF1H = COEFN1 +
COEF31*TERMO1
COEF2V = COEFN2 -
COEF32*TERMO2
16
COEF2H = COEFN2 +
COEF32*TERMO2
COEF3V = COEFN3 -
COEF33*TERMO3
COEF3H = COEFN3 +
COEF33*TERMO3
17
COEF4V = COEFN4 -
COEF34*TERMO4
COEF4H = COEFN4 +
COEF34*TERMO4
COEF5V = COEFN4 -
COEF44*TERMO4
18
COEF5H = COEFN4 +
COEF44*TERMO4
23
TAU1 =
XKO*A1*CA1PTS
TAU2 =
XKO*A2*CA1PTS
TAU3 =
XKO*A2*CA2PTS
TAU4 =
XKO*A4*CA2PTS
24
FTAU1 = F(TAU1)
FTAU2 = F(TAU2)
FTAU3 = F(TAU3)
FTAU4 = F(TAU4)
25
SI1 =
2.0*XKO*A1*STHT
SI2 =
2.0*XKO*A2*STHT
SI4 =
2.0*XKO*A4*STHT
26
7.01
BESL
(SI1,XJOX1,
XJ1X1,XJ2X1)
H
H
H
H
H
/31.01

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI)



2



```
TERM4P =
CMPLX(XJOX4,
XJ1X4)

TERM4M =
CONJG(TERM4P)

TERM5 = XJOX1 +
XJ2X1

05
TERM6 = XJOX2 +
XJ2X2

TERM7 = XJOX4 +
XJ2X4

06
PHI1 =
XKO2C*(H1 + H2)

PHI2 = XKO2C*(H2)

PHI3 = PHI2

PHI4 =
XKO2C*(H2 + H3)

07
PHASE1 =
CMPLX(COS(PHI1),
- SIN(PHI1))

PHASE2 =
CMPLX(COS(PHI2),
- SIN(PHI2))

08
PHASE3 =
CMPLX(COS(PHI3),
SIN(PHI3))

PHASE4 =
CMPLX(COS(PHI4),
SIN(PHI4))
```

```
FTAU8 = F(TAU8)

12
FFVV01 =
A1*TERM1M*COEF1V*
PHASE1

FFHH01 =
A1*TERM1M*COEF1H*
PHASE1

FFVV02 =
A2*TERM2M*COEF2V*
PHASE2

13
FFHH02 =
A2*TERM2M*COEF2H*
PHASE2

FFVV03 =
A2*TERM2M*COEF3V*
PHASE3

FFHH03 =
A2*TERM2M*COEF3H*
PHASE3

14
FFVV04 =
A4*TERM4M*COEF4V*
PHASE4

FFHH04 =
A4*TERM4M*COEF4H*
PHASE4

FFVV05 =
A1*TERM1P*COEF5V*
PHASE1
```

```
*FTAU2*PHASE2

FFHH09 = - FFVV09

18
FFVV10 =
A2*TERM2M*TANAP2*
(0.5)
*FTAU3*PHASE3

FFHH10 = - FFVV10

FFVV11 =
A4*TERM4M*TANAP2*
(-0.5)
*FTAU4*PHASE4

19
FFHH11 = - FFVV11

FFVV12 =
A1*TERM1P*TANAM1*
(0.5)
*FTAU5*PHASE1*Q6

FFHH12 = - FFVV12

20
FFVV13 =
A2*TERM2P*TANAM1*
(-0.5)
*FTAU6*PHASE2*Q6

FFHH13 = - FFVV13

21
FFVV14 =
A2*TERM2P*TANAM2*
(-.5)*(1. -
FTAU7)*PHASE3*Q8

FFHH14 = - FFVV14

/
/32.01
```

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI)

L-90

```
                                                                      ----------
                                                                     /   30    /
                                                                      ----------
                                                                          |
       31.21--->*                                              31.09--->|
                |            01                                           | NOTE  09
     *--------------------*                                    * * * * * * * * * * *
     |     FFVV15 =       |                                    *    CONTINUE       *
     | A4*TERM4P*TANAM2*  |                                    * * * * * * * * * * *
     |      (-0.5)        |                                           |
     | *FTAU5*PHASE4*Q8   |                                           |
     |                    |                                           |           10
     | FFHH15 = - FFVV15  |                                    *--------------------*
     |                    |                                    |     FFVV01 =       |
     |    FFVV16 = -      |                                    | A1*TERM1M*COEF1V*  |
     | A1*COEFN1*TERM5*   |                                    |    PHASE1*Q1       |
     |      PHASE1        |                                    |                    |
     *--------------------*                                    |     FFHH01 =       |
                |                                              | A1*TERM1M*COEF1H*  |
                |                                              |    PHASE1*Q1       |
                |            02                                |                    |
     *--------------------*                                    |     FFVV02 =       |
     | FFHH16 = FFVV16    |                                    | A2*TERM2M*COEF2V*  |
     |                    |                                    |    PHASE2*Q2       |
     |    FFVV17 = -      |                                    *--------------------*
     | A2*COEFN2*TERM6*   |                                           |
     |      PHASE2        |                                           |
     |                    |                                           |           11
     | FFHH17 = FFVV17    |                                    *--------------------*
     *--------------------*                                    |     FFHH02 =       |
                |                                              | A2*TERM2M*COEF2H*  |
                |                                              |    PHASE2*Q2       |
                |            03                                |                    |
     *--------------------*                                    |     FFVV03 =       |
     |    FFVV18 = -      |                                    | A2*TERM2M*COEF3V*  |
     | A2*COEFN3*TERM6*   |                                    |    PHASE3*Q3       |
     |      PHASE3        |                                    |                    |
     |                    |                                    |     FFHH03 =       |
     | FFHH18 = FFVV18    |                                    | A2*TERM2M*COEF3H*  |
     |                    |                                    |    PHASE3*Q3       |
     |    FFVV19 = -      |                                    *--------------------*
     | A4*COEFN4*TERM7*   |                                           |
     |      PHASE4        |                                           |
     *--------------------*                                           |           12
                |                                              *--------------------*
                |                                              |     FFVV04 =       |
                |            04                                | A4*TERM4M*COEF4V*  |
     *--------------------*
```

2

```
           FFVV19 =
*-------------------------*
| A4*COEFN4*TERM7*        |
|       PHASE4            |
*-------------------------*
             |
             |
             |                 04
*-------------------------*
|  FFHH19 = FFVV19        |
|                         |
|     FFVV20 = -          |
| A1*COEFN1*TERM5*        |
|       PHASE1            |
|                         |
|  FFHH20 = FFVV20        |
*-------------------------*
             |
             |                 05
*-------------------------*
|     FFVV21 = -          |
| A2*COEFN2*TERM6*        |
|    PHASE2*Q6            |
|                         |
|  FFHH21 = FFVV21        |
|                         |
|     FFVV22 = -          |
| A4*COEFN4*TERM7*        |
|    PHASE4*Q8            |
*-------------------------*
             |
             |                 06
*-------------------------*
|  FFHH22 = FFVV22        |
*-------------------------*
             |
             |                 07
*-------------------------*
|  FFVV = FFVV01 +        |
| FFVV02 + FFVV03 +       |
| FFVV04 + FFVV05 +       |
| FFVV06 + FFVV07 +       |
| FFVV08 + FFVV09 +       |
| FFVV10 + FFVV11 +       |
| FFVV12 + FFVV13 +       |
| FFVV14 + FFVV15 +       |
| FFVV16 + FFVV17 +       |
| FFVV18 + FFVV19 +       |
| FFVV20 + FFVV21 +       |
|       FFVV22            |
*-------------------------*
             |
             +------------------+
```

```
+---------------->*
                  |              08
*-----------------------*
|   FFHH = FFHH01 +     |
| FFHH02 + FFHH03 +     |
| FFHH04 + FFHH05 +     |
| FFHH06 + FFHH07 +     |
| FFHH08 + FFHH09 +     |
| FFHH10 + FFHH11 +     |
| FFHH12 + FFHH13 +     |
| FFHH14 + FFHH15 +     |
| FFHH16 + FFHH17 +     |
| FFHH18 + FFHH19 +     |
| FFHH20 + FFHH21 +     |
```

```
*--------------------------*
             |
             |                 12
*--------------------------*
|        FFVV04 =          |
|  A4*TERM4M*COEF4V*       |
|         PHASE4           |
|                          |
|        FFHH04 =          |
|  A4*TERM4M*COEF4H*       |
|         PHASE4           |
|                          |
|        FFVV05 =          |
|  A4*TERM4P*COEF5V*       |
|         PHASE4           |
*--------------------------*
             |
             |                 13
*--------------------------*
|        FFHH05 =          |
|  A4*TERM4P*COEF5H*       |
|         PHASE4           |
|                          |
|        FFVV06 =          |
|  A1*TERM1M*TANAP1*       |
|          (0.5)           |
|  *FTAU1*PHASE1*Q1        |
|                          |
|  FFHH06 = - FFVV06       |
*--------------------------*
             |
             |                 14
*--------------------------*
|        FFVV07 =          |
|  A2*TERM2M*TANAP1*       |
|         (-0.5)           |
|  *FTAU2*PHASE2*Q1        |
|                          |
|  FFHH07 = - FFVV07       |
|                          |
|        FFVV08 =          |
|  A2*TERM2M*TANAP2*       |
|          (0.5)           |
|  *FTAU3*PHASE3*Q3        |
*--------------------------*
             |
             |                 15
*--------------------------*
|  FFHH08 = - FFVV08       |
|                          |
|        FFVV09 =          |
|  A2*TERM2M*TANATT*       |
|          (0.5)           |
|  *PHASE3*(Q2 - Q3)       |
|                          |
|  FFHH09 = - FFVV09       |
*--------------------------*
             |
                                                              |      FFVV06 =       |
                                                              | A1*TERM1M*TANAP1*   |
                                                              |       (0.5)         |
                                                              | *FTAU1*PHASE1*Q1    |
                          |                                   |                     |
                          |                                   | FFHH06 = - FFVV06   |
        *-----------------*                                   *---------------------*
               |                                                         |
               |         C6                                              |
        *-----------------*                                              |       14
        | FFHH22 = FFVV22 |                                   *---------------------*
        *-----------------*                                   |      FFVV07 =       |
               |                                              | A2*TERM2M*TANAP1*   |
               |         07                                   |       (-0.5)        |
        *-----------------------*                             | *FTAU2*PHASE2*Q1    |
        |  FFVV = FFVV01 +      |                             |                     |
        | FFVV02 + FFVV03 +     |                             | FFHH07 = - FFVV07   |
        | FFVV04 + FFVV05 +     |                             |                     |
        | FFVV06 + FFVV07 +     |                             |      FFVV08 =       |
        | FFVV08 + FFVV09 +     |                             | A2*TERM2M*TANAP2*   |
        | FFVV10 + FFVV11 +     |                             |       (0.5)         |
        | FFVV12 + FFVV13 +     |                             | *FTAU3*PHASE3*Q3    |
        | FFVV14 + FFVV15 +     |                             *---------------------*
        | FFVV16 + FFVV17 +     |                                        |
        | FFVV18 + FFVV19 +     |  +----------------->*                  |
        | FFVV20 + FFVV21 +     |  |                  |       08         |       15
        |      FFVV22           |  |       *--------------------*   *---------------------*
        *-----------------------*  |       |  FFHH = FFHH01 +   |   | FFHH08 = - FFVV08   |
               |                   |       | FFHH02 + FFHH03 +  |   |                     |
               |                   |       | FFHH04 + FFHH05 +  |   |      FFVV09 =       |
               +-------------------+       | FFHH06 + FFHH07 +  |   | A2*TERM2M*TANATT*   |
                                           | FFHH08 + FFHH09 +  |   |       (0.5)         |
                                           | FFHH10 + FFHH11 +  |   | *PHASE3*(Q2 - Q3)   |
                                           | FFHH12 + FFHH13 +  |   |                     |
                                           | FFHH14 + FFHH15 +  |   | FFHH09 = - FFVV09   |
                                           | FFHH16 + FFHH17 +  |   *---------------------*
                                           | FFHH18 + FFHH19 +  |             |
                                           | FFHH20 + FFHH21 +  |             |
                                           |      FFHH22        |             |  /
                                           *--------------------*           -----
                                                     |                     /33.01
                                                     |
                                                     |
                                                    ...
                                                   .   .
                                                  .33.07.
                                                   .   .
                                                    ... 40
```

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI)

```
                    32.15--->*
                             |                01
                    *----------------------*
                    |     FFVV11 =         |
                    | A4*TERM4M*TANAP2*    |
                    |       (-0.5)         |
                    | *FTAU4*PHASE4*Q3     |
                    |                      |
                    | FFHH11 = - FFVV11    |
                    |                      |
                    |     FFVV12 = -       |
                    | A1*COEFN1*TERM5*     |
                    |     PHASE1*Q1        |
                    *----------------------*
                             |
                             |
                             |                02
                    *----------------------*
                    |  FFHH12 = FFVV12     |
                    |                      |
                    |     FFVV13 = -       |
                    | A2*COEFN2*TERM6*     |
                    |     PHASE2*Q2        |
                    |                      |
                    |  FFHH13 = FFVV13     |
                    *----------------------*
                             |
                             |
                             |                03
                    *----------------------*
                    |     FFVV14 = -       |
                    | A2*COEFN3*TERM6*     |
                    |     PHASE3*Q3        |
                    |                      |
                    |  FFHH14 = FFVV14     |
                    |                      |
                    |     FFVV15 = -       |
```

2

```
*-----------------------*
|   FFVV14 = -          |
| A2*COEFN3*TERM6*      |
|     PHASE3*C3         |
|                       |
|   FFHH14 = FFVV14     |
|                       |
|   FFVV15 = -          |
| A4*COEFN4*TERM7*      |
|     PHASE4*Q8         |
*-----------------------*
            |
            |
            |         04
*-----------------------*
|  FFHH15 = FFVV15      |
|                       |
|   FFVV16 = -          |
| A4*COEFN4*TERM7*      |
|      PHASE4           |
|                       |
|  FFHH16 = FFVV16      |
*-----------------------*
            |
            |
            |         05
*-----------------------*
|  FFVV = FFVV01 +      |
| FFVV02 + FFVV03 +     |
| FFVV04 + FFVV05 +     |
| FFVV06 + FFVV07 +     |
| FFVV08 + FFVV09 +     |
| FFVV11 + FFVV12 +     |
| FFVV13 + FFVV14 +     |    +---------------------->*
|  FFVV15 + FFVV16      |    |                       |           08
*-----------------------*    |           *-----------------------*
            |                |           |      CFVV = -         |
            |                |           |    CONJG(FFVV)        |
            |         06     |           |                       |
*-----------------------*    |           |      CFHH =           |
|  FFHH = FFHH01 +      |    |           |    CONJG(FFHH)        |
| FFHH02 + FFHH03 +     |    |           *-----------------------*
| FFHH04 + FFHH05 +     |    |                       |
| FFHH06 + FFHH07 +     |    |                       |
| FFHH08 + FFHH09 +     |    |                       |           09
| FFHH11 + FFHH12 +     |    |           *-----------------------*
| FFHH13 + FFHH14 +     |    |           |     EVVR(I) =         |
|  FFHH15 + FFHH16      |    |           |    REAL(CFVV)         |
*-----------------------*    |           |                       |
            |                |           |     EVVI(I) =         |
  32.08--->|                |           |    AIMAG(CFVV)        |
    40      |         07     |           |                       |
*-----------------------*    |           |     EHHR(I) =         |
|      FFVV =           |    |           |    REAL(CFHH)         |
| FFVV*.02539998*       |    |           |                       |
|       SPI             |    |           |     EHHI(I) =
*-----------------------*
| FFVV = FFVV01 +       |
| FFVV02 + FFVV03 +     |
| FFVV04 + FFVV05 +     |
| FFVV06 + FFVV07 +     |
| FFVV08 + FFVV09 +     |
| FFVV11 + FFVV12 +     |
| FFVV13 + FFVV14 +     |           +------------------>*
|  FFVV15 + FFVV16      |           |                   |        08
*-----------------------*           |            *-------------------*
            |                       |            |   CFVV = -        |
            |                       |            |   CONJG(FFVV)     |
            |           06          |            |                   |
*-----------------------*           |            |   CFHH =          |
| FFHH = FFHH01 +       |           |            |   CONJG(FFHH)     |
| FFHH02 + FFHH03 +     |           |            *-------------------*
| FFHH04 + FFHH05 +     |           |                   |
| FFHH06 + FFHH07 +     |           |                   |
| FFHH08 + FFHH09 +     |           |                   |        09
| FFHH11 + FFHH12 +     |           |            *-------------------*
| FFHH13 + FFHH14 +     |           |            |   EVVR(I) =       |
|  FFHH15 + FFHH16      |           |            |   REAL(CFVV)      |
*-----------------------*           |            |                   |
            |                       |            |   EVVI(I) =       |
  32.08---->|                       |            |   AIMAG(CFVV)     |
    40      |           07          |            |                   |
*-----------------------*           |            |   EHHR(I) =       |
|       FFVV =          |           |            |   REAL(CFHH)      |
| FFVV*.02539998*       |           |            |                   |
|        SPI            |           |            |   EHHI(I) =       |
|                       |           |            |   AIMAG(CFHH)     |
|       FFHH =          |           |            *-------------------*
| FFHH*.02539998*       |           |                   |
|        SPI            |           |                   |
*-----------------------*           |                   |
            |                       |          900      *    10
            |                       |                *     *
            +-----------------------+             *           *
                                                *               *    NO
                                              *    END OF DO     *--+
                                                *   LOOP?       *   |
                                                  *           *     |
                                                     *     *        |
                                                        *         ....
                                                        |YES     . 30 .
                                                        |        . 22 .
                                                        |         ....
                                                        |
                                                        |
                                                        | 11
                                                    ---------
                                                    * EXIT  *
                                                    ---------
```

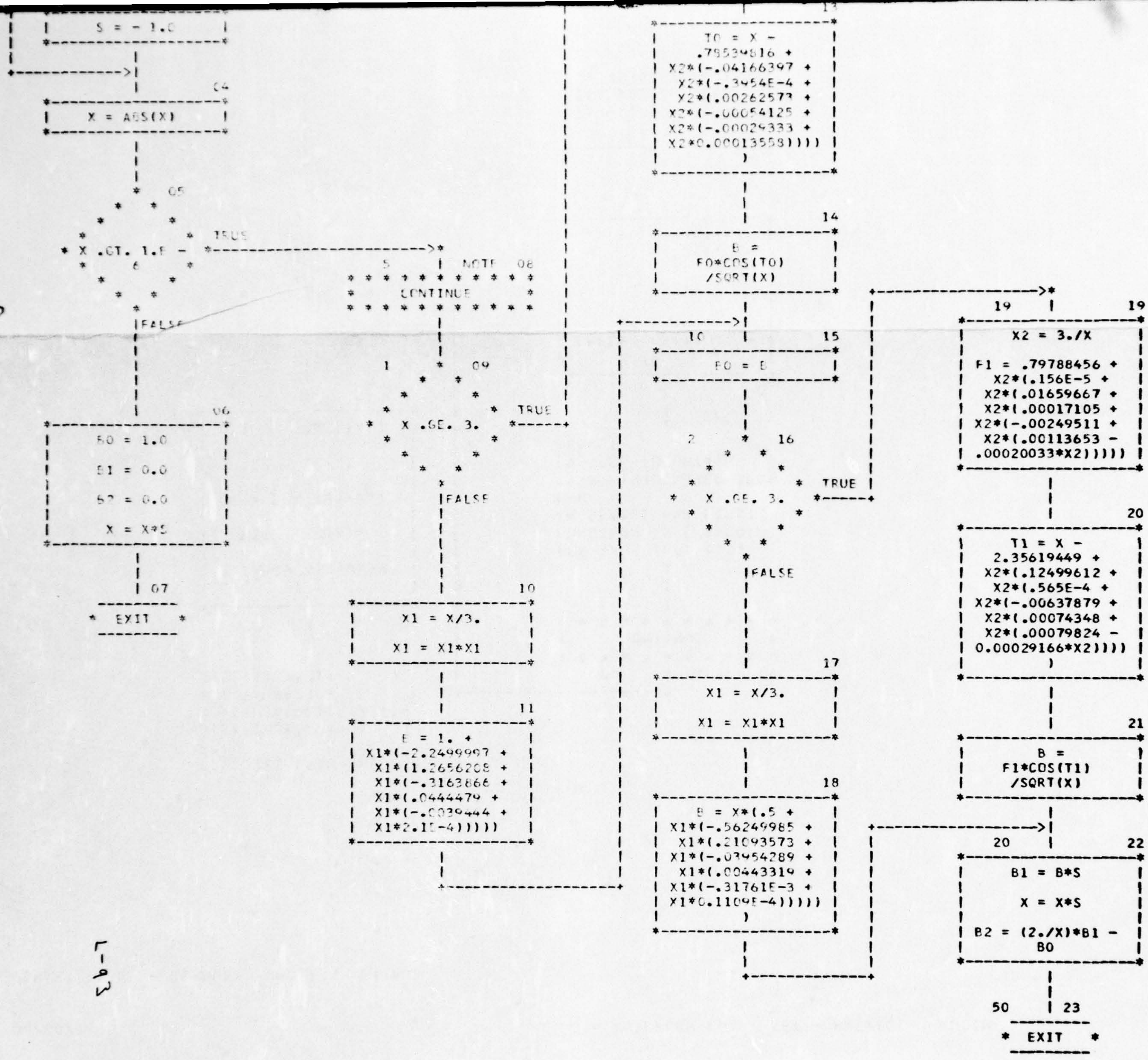
CHART TITLE - NON-PROCEDURAL STATEMENTS

```
             COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, TO
             COMPLEX TERM1P, TERM1M, TERM2P, TERM2M, TERM4P, TERM4M, PHASE1,
              PHASE2,PHASE3, PHASE4, FFVV01, FFVV02, FFVV03, FFVV04, FFVV05,
              FFVV06,FFVV07, FFVV08, FFVV09, FFVV10, FFVV11, FFVV12, FFVV13,
              FFVV14,FFVV15, FFVV16, FFVV17, FFVV18, FFVV19, FFVV20, FFVV21,
              FFVV22,FFHH01, FFHH02, FFHH03, FFHH04, FFHH05, FFHH06, FFHH07,
              FFHH08,FFHH09, FFHH10, FFHH11, FFHH12, FFHH13, FFHH14, FFHH15,
              FFHH16,FFHH17, FFHH18, FFHH19, FFHH20, FFHH21, FFHH22, FFVV,
              FFHH,  CFVV,   CFHH,   FTAU1,  FTAU2,  FTAU3,  FTAU4,  FTAU5,
              FTAU6, FTAU7,  FTAU8
             COMPLEX F
             DIMENSION FVVR(512), FVVI(512), EHHR(512), EHHI(512)
1000         FORMAT(9F7.2,I2)
1010         FORMAT ( 31H1 ' FRUSTRA-CYLINDER-FRUSTRA  ',/
                      30H ' UFIMTSEV-RUCK FORMULATION ',/
                       8H   A1 = ,F8.3, 8H   H1 = ,F8.3,/
                       8H   A2 = ,F8.3, 8H   H2 = ,F8.3,/
                       8H   A4 = ,F8.3, 8H   H3 = ,F8.3,/)
2010         FORMAT ( 18H0  ASPECT ANGLE = , F8.3,/,
                      11H  ALPHA1 = ,F8.3,/, 11H  ALPHA2 = ,F8.3,/,
                      19H  SHADOW(ALPHA3) = ,F8.3 )
```

CHART TITLE - SUBROUTINE BESL(X,F0,B1,B2)

```
      ------------
     /   BESL    /
     ------------

           *

   * BESSEL FUNCTION
  SUBROUTINE UTILIZING
  POLYNOMIAL
  APPROXIMATIONS
   * COMPUTES J0,J1,OR
  J2 FOR POSITIVE REAL
  ARGUMENTS
   * REFERENCE (HNDBK
  MATH FUNCT BY
  ABRAMOWITZ AND STEGUN
  SECTION 9.4 )

           *
           |              01
   *-------------------------*
   |       S = 1.0           |
   *-------------------------*
           |
           |
           |
           *   02
         *   *
       *       *
 FALSE *           *
 +-----*  X .LT. 0.0  *
 |     *           *
 |       *       *
 |         *   *
 |           *
 |           |TRUE
 |           |
 |           |
 |           |
 |           |              03
 |   *-------------------------*
 |   |      S = - 1.0          |
 |   *-------------------------*
 |           |
 +---------->|
             |              04
     *-------------------------*
     |      X = ABS(X)         |
     *-------------------------*
             |
             |
             *   05
           *   *
         *       *
       *           *   TRUE
```

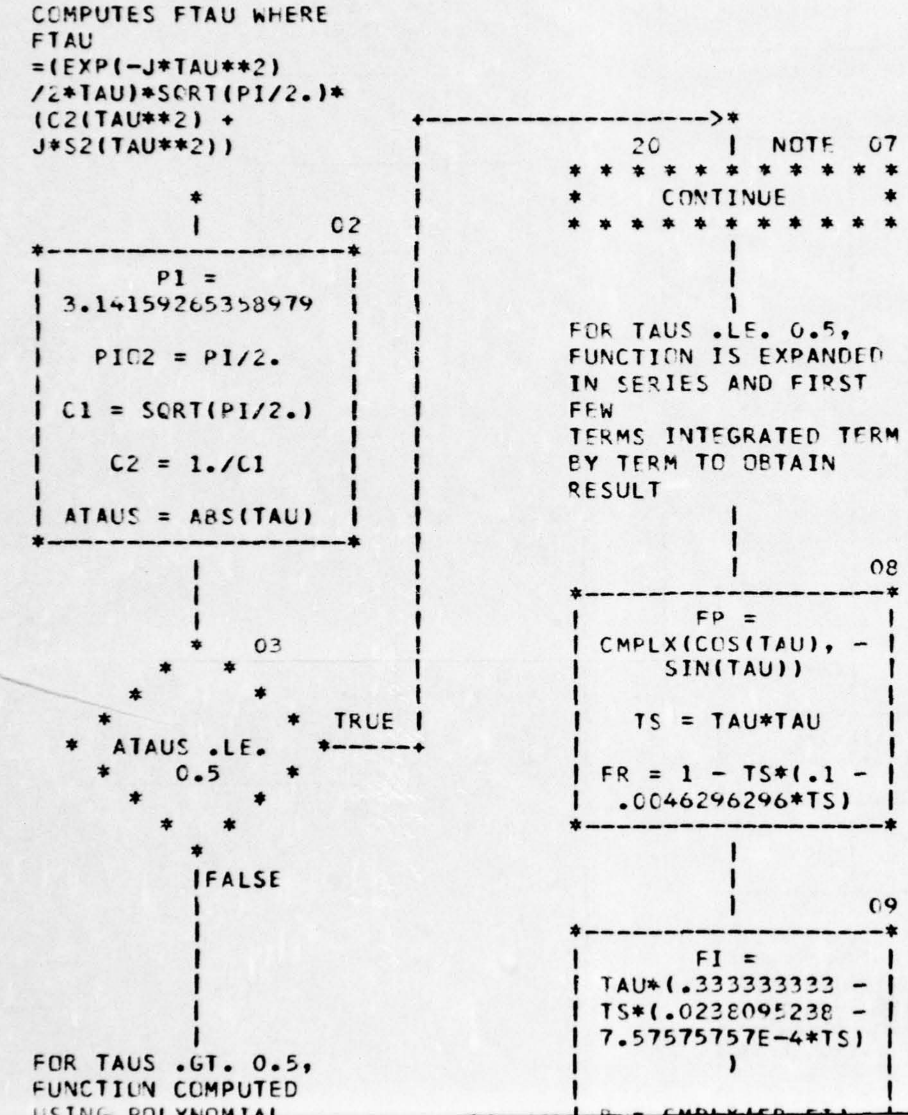
```
+---------------------->*
|            9          |              12
|         *------------------------*
|         |       X2 = 3./X        |
|         |                        |
|         | F0 = .79788456 +       |
|         |   X2*(-.77E-6 +        |
|         | X2*(-.00552740 +       |
|         |  X2*(-.9512E-4 +       |
|         |  X2*(.00137237 +       |
|         | X2*(-.72805E-3 +       |
|         | X2*0.14476E-3))))      |
|         |           )            |
|         *------------------------*
|                       |
|                       |
|                       |              13
|         *------------------------*
|         |      T0 = X -          |
|         |     .78539816 +        |
|         | X2*(-.04166397 +       |
|         |  X2*(-.3954E-4 +       |
|         |  X2*(.00262573 +       |
|         | X2*(-.00054125 +       |
|         | X2*(-.00029333 +       |
|         | X2*0.00013558))))      |
|         |           )            |
|         *------------------------*
|                       |
|                       |
|                       |              14
|         *------------------------*
```


S = - 1.0
04
X = ABS(X)
05
X .GT. 1.E - 6
TRUE
FALSE
06
B0 = 1.0
B1 = 0.0
B2 = 0.0
X = X*S
07
EXIT
5
NOTE 08
CONTINUE
1
09
X .GE. 3.
TRUE
FALSE
10
X1 = X/3.
X1 = X1*X1
11
B = 1. +
X1*(-2.2499997 +
X1*(1.2656208 +
X1*(-.3163866 +
X1*(.0444479 +
X1*(-.0039444 +
X1*2.1E-4)))))
13
T0 = X -
.78539816 +
X2*(-.04166397 +
X2*(-.3954E-4 +
X2*(.00262573 +
X2*(-.00054125 +
X2*(-.00029333 +
X2*0.00013558))))
)
14
B =
F0*COS(T0)
/SQRT(X)
10
15
F0 = B
2
16
X .GE. 3.
TRUE
FALSE
17
X1 = X/3.
X1 = X1*X1
18
B = X*(.5 +
X1*(-.56249985 +
X1*(.21093573 +
X1*(-.03954289 +
X1*(.00443319 +
X1*(-.31761E-3 +
X1*0.1109E-4))))))
)
19
19
X2 = 3./X
F1 = .79788456 +
X2*(.156E-5 +
X2*(.01659667 +
X2*(.00017105 +
X2*(-.00249511 +
X2*(.00113653 -
.00020033*X2)))))
20
T1 = X -
2.35619449 +
X2*(.12499612 +
X2*(.565E-4 +
X2*(-.00637879 +
X2*(.00074348 +
X2*(.00079824 -
0.00029166*X2))))
)
21
B =
F1*COS(T1)
/SQRT(X)
20
22
B1 = B*S
X = X*S
B2 = (2./X)*B1 -
B0
50
23
EXIT

2

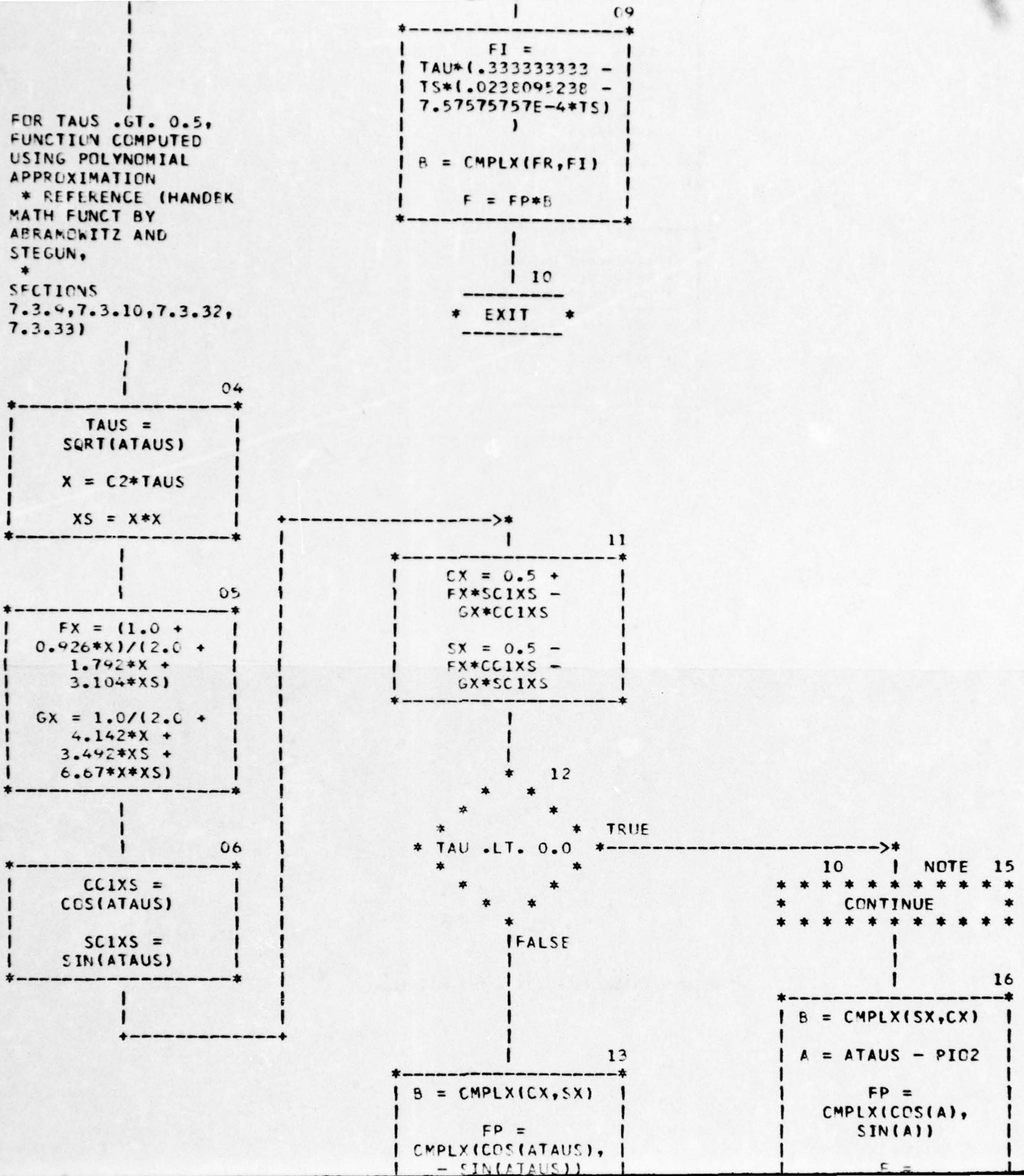
CHART TITLE - COMPLEX FUNCTION F(TAU)

L-94



```
COMPUTES FTAU WHERE
FTAU
=(EXP(-J*TAU**2)
/2*TAU)*SQRT(PI/2.)*
(C2(TAU**2) +                   +-------------------->*
J*S2(TAU**2))                   |            20     | NOTE  07
                                |            * * * * * * * * * * *
             *                  |            *     CONTINUE       *
             |          02      |            * * * * * * * * * * *
*-----------------------*       |                   |
|         PI =          |       |                   |
| 3.14159265358979      |       |                   |
|                       |       |            FOR TAUS .LE. 0.5,
|     PIO2 = PI/2.      |       |            FUNCTION IS EXPANDED
|                       |       |            IN SERIES AND FIRST
| C1 = SQRT(PI/2.)      |       |            FEW
|                       |       |            TERMS INTEGRATED TERM
|     C2 = 1./C1        |       |            BY TERM TO OBTAIN
|                       |       |            RESULT
| ATAUS = ABS(TAU)      |       |                   |
*-----------------------*       |                   |
             |                  |                   |         08
             |                  |            *-------------------*
             *    03            |            |       FP =        |
          *     *               |            | CMPLX(COS(TAU), - |
       *           *            |            |     SIN(TAU))     |
     *               *  TRUE    |            |                   |
   *  ATAUS .LE.       *--------+            |    TS = TAU*TAU   |
     *    0.5        *                       |                   |
       *           *                         | FR = 1 - TS*(.1 - |
          *     *                            |  .0046296296*TS)  |
             *                               *-------------------*
             |FALSE                                 |
             |                                      |
             |                                      |         09
             |                               *-------------------*
             |                               |       FI =        |
             |                               | TAU*(.333333333 - |
             |                               | TS*(.0238095238 - |
             |                               | 7.57575757E-4*TS) |
FOR TAUS .GT. 0.5,                           |         )         |
FUNCTION COMPUTED                            |                   |
USING POLYNOMIAL
```

2



```
FOR TAUS .GT. 0.5,
FUNCTIUN COMPUTED
USING POLYNOMIAL
APPROXIMATION
 * REFERENCE (HANDBK
MATH FUNCT BY
ABRAMOWITZ AND
STEGUN,
 *
SECTIONS
7.3.9,7.3.10,7.3.32,
7.3.33)

04
TAUS =
SQRT(ATAUS)

X = C2*TAUS

XS = X*X

05
FX = (1.0 +
0.926*X)/(2.0 +
1.792*X +
3.104*XS)

GX = 1.0/(2.0 +
4.142*X +
3.492*XS +
6.67*X*XS)

06
CC1XS =
COS(ATAUS)

SC1XS =
SIN(ATAUS)

09
FI =
TAU*(.333333333 -
TS*(.0238095238 -
7.57575757E-4*TS)
)

B = CMPLX(FR,FI)

F = FP*B

10
* EXIT *

11
CX = 0.5 +
FX*SC1XS -
GX*CC1XS

SX = 0.5 -
FX*CC1XS -
GX*SC1XS

12
* TAU .LT. 0.0 *
TRUE
FALSE

10   NOTE 15
CONTINUE

13
B = CMPLX(CX,SX)

FP =
CMPLX(COS(ATAUS),
- SIN(ATAUS))

16
B = CMPLX(SX,CX)

A = ATAUS - PIO2

FP =
CMPLX(COS(A),
SIN(A))

F =
```

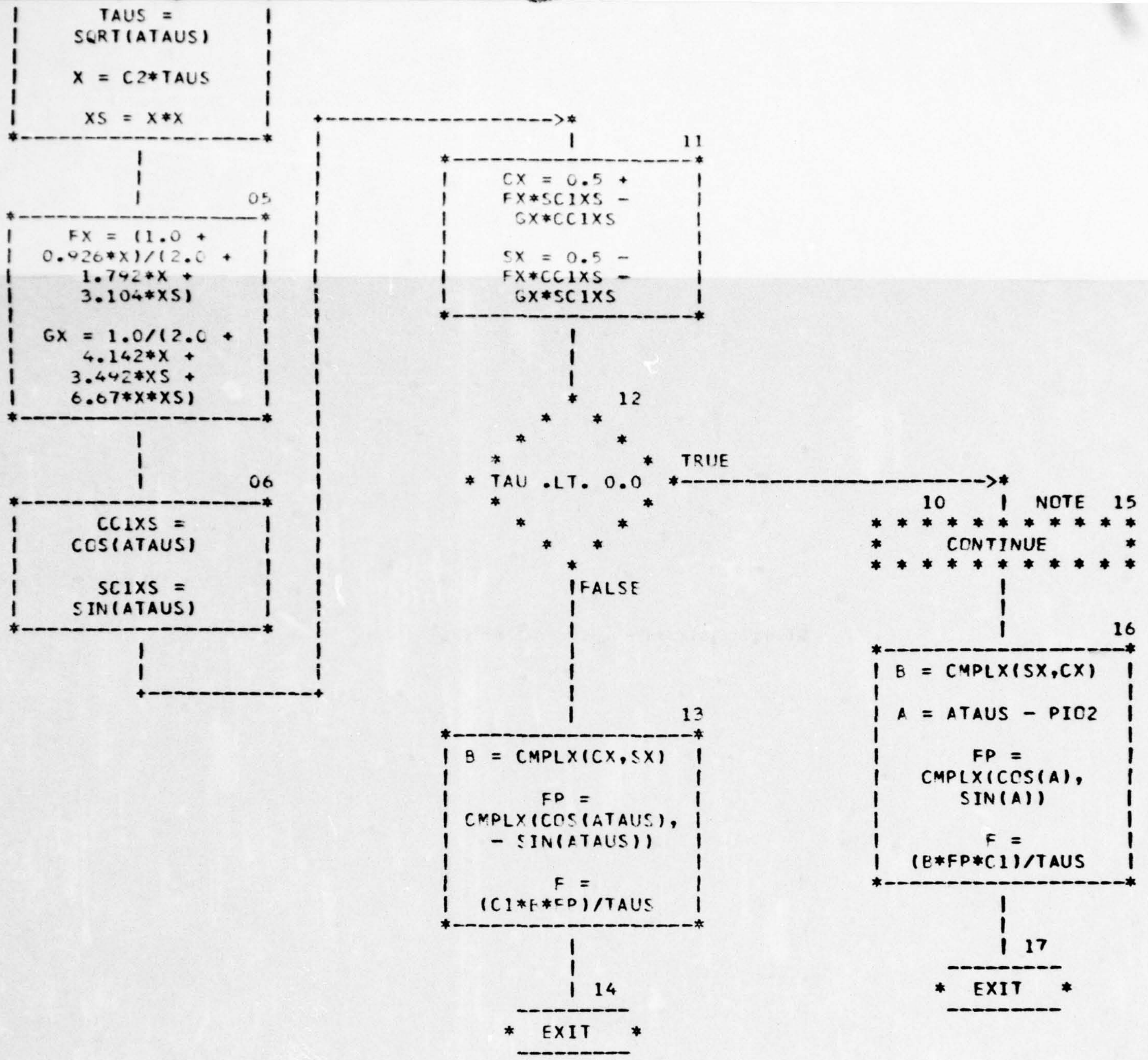

TAUS =
SQRT(ATAUS)
X = C2*TAUS
XS = X*X
05
FX = (1.0 +
0.926*X)/(2.0 +
1.792*X +
3.104*XS)
GX = 1.0/(2.0 +
4.142*X +
3.442*XS +
6.67*X*XS)
06
CC1XS =
COS(ATAUS)
SC1XS =
SIN(ATAUS)
11
CX = 0.5 +
FX*SC1XS -
GX*CC1XS
SX = 0.5 -
FX*CC1XS -
GX*SC1XS
12
TAU .LT. 0.0
TRUE
FALSE
13
B = CMPLX(CX,SX)
FP =
CMPLX(COS(ATAUS),
- SIN(ATAUS))
F =
(C1*B*FP)/TAUS
14
EXIT
10
NOTE 15
CONTINUE
16
B = CMPLX(SX,CX)
A = ATAUS - PIO2
FP =
CMPLX(COS(A),
SIN(A))
F =
(B*FP*C1)/TAUS
17
EXIT

CHART TITLE - NON-PROCEDURAL STATEMENTS

COMPLEX B, FP

AUTOFLOW CHART SET - FWO/SCL RADSIM

ILF - NON-PROCFDURAL STATEMENTS

COMPLEX B, FP

L-96

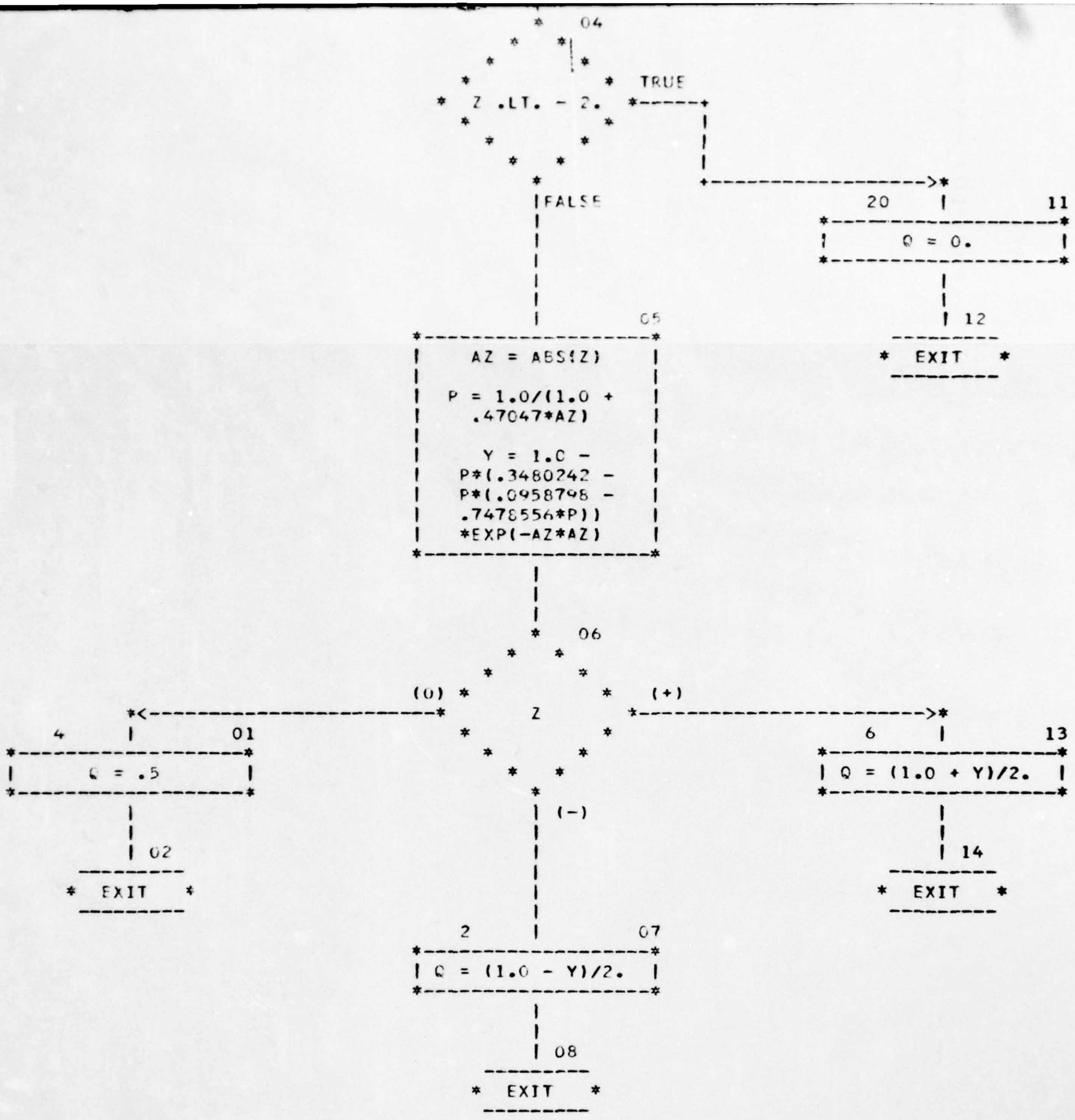
2

CHART TITLE - FUNCTION Q(Z)

L-96

```
                                 Q(Z) = 0.5*(1 +
                                 ERF(Z))
                                  * ERF(Z) IS
                                 EVALUATED USING A
                                 RATIONAL POLYNOMIAL
                                 APPROXIMATION
                                  * REFERENCE (HANDBK
                                 MATH FUNCT BY
                                 ABRAMOWITZ AND
                                 STEGUN,
                                  *          SECTION
                                 7.1.26)

                                       *
                                       I
                                       *   03
                                     *   *
                                   *       *
                                 *           *  TRUE
                               *  Z .GT. 2.   *------------------->*
                                 *           *                10   I          09
                                   *       *                  *--------------------*
                                     *   *                    I       Q = 1.       I
                                       *                      *--------------------*
                                       IFALSE                          I
                                       I                               I
                                       I                               I 10
                                       I                            ----------
                                       I                          *   EXIT   *
                                       I                            ----------
                                       *   04
                                     *   *
                                   *       *
                                 *           *  TRUE
                               * Z .LT. - 2.  *-----+
                                 *           *      I
                                   *       *        I
                                     *   *          I
                                       *            +-------------->*
```

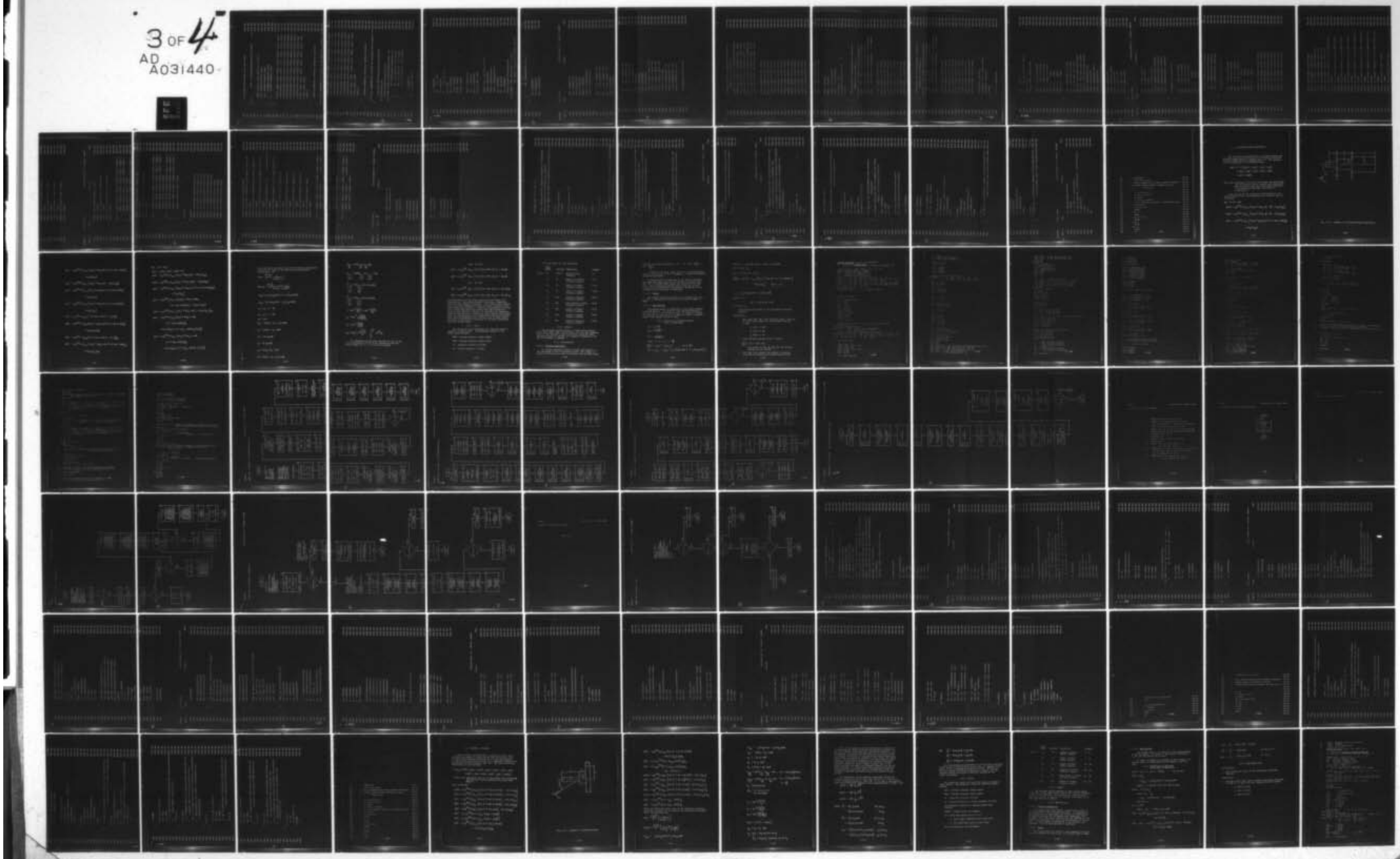
2


04
Z .LT. - 2.
TRUE
FALSE
20
11
Q = 0.
12
EXIT
05
AZ = ABS(Z)
P = 1.0/(1.0 + .47047*AZ)
Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P)) *EXP(-AZ*AZ)
06
Z
(0)
(+)
(-)
4
01
Q = .5
02
EXIT
6
13
Q = (1.0 + Y)/2.
14
EXIT
2
07
Q = (1.0 - Y)/2.
08
EXIT

AD-A031 440   GENERAL DYNAMICS FORT WORTH TEX CONVAIR AEROSPACE DIV   F/G 17/9
ENDO ATMOSPHERIC EXO ATMOSPHERIC RADAR MODELING. RADAR CROSS SE--ETC(U)
JUN 76 R J HANCOCK, F H CLEVELAND   F30602-73-C-0380

UNCLASSIFIED   RADC-TR-76-186-VOL-4-PT-2   NL

3 OF 4
AD
A031440



```
      SUBROUTINE TARGET (EVVR, EVVI, EHHR, EHHI)                        RCS7 001
C                                                                       RCS7 002
C * * *          ST-3A -- FRUSTRA-CYLINDER-FRUSTRA (UFIMTSEV)      * * * RCS7 003
C                                                                       RCS7 004
      COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, TO                       RCS7 005
C     NMIN = MINIMUM FREQUENCY SAMPLE                                   RCS7 006
C     NMAX = MAXIMUM FREQUENCY SAMPLE                                   RCS7 007
C     LF   = FREQUENCY INCREMENT IN MHZ                                 RCS7 008
C     FC   = CARRIER FREQUENCY IN GHZ                                   RCS7 009
C                                                                       RCS7 010
      COMPLEX TERM1P, TERM1M, TERM2P, TERM2M, TERM4P, TERM4M, PHASE1,   RCS7 011
     1 PHASE2,PHASE3, PHASE4, FFVV01, FFVV02, FFVV03, FFVV04, FFVV05,   RCS7 012
     2 FFVV06,FFVV07, FFVV08, FFVV09, FFVV10, FFVV11, FFVV12, FFVV13,   RCS7 013
     3 FFVV14,FFVV15, FFVV16, FFVV17, FFVV18, FFVV19, FFVV20, FFVV21,   RCS7 014
     4 FFVV22,FFHH01, FFHH02, FFHH03, FFHH04, FFHH05, FFHH06, FFHH07,   RCS7 015
     5 FFHH08,FFHH09, FFHH10, FFHH11, FFHH12, FFHH13, FFHH14, FFHH15,   RCS7 016
     6 FFHH16,FFHH17, FFHH18, FFHH19, FFHH20, FFHH21, FFHH22, FFVV,     RCS7 017
     7 FFHH,  CFVV,   CFHH,   FTAU1,  FTAU2,  FTAU3,  FTAU4,  FTAU5,    RCS7 018
     8 FTAU6, FTAU7,  FTAU8                                             RCS7 019
      COMPLEX F                                                         RCS7 020
C                                                                       RCS7 021
      DIMENSION EVVR(512), EVVI(512), EHHR(512), EHHI(512)              RCS7 022
C                                                                       RCS7 023
C * * * ALL DIMENSIONS ARE IN INCHES AND ALL ANGLES ARE IN DEGREES  * * RCS7 024
```

```
      COMPLEX TERM1P, TERM1M, TERM2P, TERM2M, TERM4P, TERM4M, PHASE1,   RCS7 011
     1 PHASE2,PHASE3, PHASE4, FFVV01, FFVV02, FFVV03, FFVV04, FFVV05,   RCS7 012
     2 FFVV06,FFVV07, FFVV08, FFVV09, FFVV10, FFVV11, FFVV12, FFVV13,   RCS7 013
     3 FFVV14,FFVV15, FFVV16, FFVV17, FFVV18, FFVV19, FFVV20, FFVV21,   RCS7 014
     4 FFVV22,FFHH01, FFHH02, FFHH03, FFHH04, FFHH05, FFHH06, FFHH07,   RCS7 015
     5 FFHH08,FFHH09, FFHH10, FFHH11, FFHH12, FFHH13, FFHH14, FFHH15,   RCS7 016
     6 FFHH16,FFHH17, FFHH18, FFHH19, FFHH20, FFHH21, FFHH22, FFVV,     RCS7 017
     7 FFHH,  CFVV,   CFHH,   FTAU1,  FTAU2,  FTAU3,  FTAU4,  FTAU5,    RCS7 018
     8 FTAU6, FTAU7,  FTAU8                                             RCS7 019
      COMPLEX F                                                         RCS7 020
C                                                                       RCS7 021
      DIMENSION EVVR(512), EVVI(512), EHHR(512), EHHI(512)              RCS7 022
C                                                                       RCS7 023
C * * * ALL DIMENSIONS ARE IN INCHES AND ALL ANGLES ARE IN DEGREES  * * RCS7 024
C                                                                       RCS7 025
      READ(5,1000) ASPECT,A1,A2,A4,H1,H2,H3,H4,H5,M1                    RCS7 026
 1000 FORMAT(9F7.2,I2)                                                  RCS7 028
      WRITE (6,1010) A1,H1,A2,H2,A4,H3                                  RCS7 029
 1010 FORMAT ( 31H1 ' FRUSTRA-CYLINDER-FRUSTRA  ',/                     RCS7 030
     1           30H ' UFIMTSEV-RUCK FORMULATION ',/                    RCS7 031
     2            8H    A1 = ,F8.3, 8H    H1 = ,F8.3,/                  RCS7 032
     3            8H    A2 = ,F8.3, 8H    H2 = ,F8.3,/                  RCS7 033
     4            8H    A4 = ,F8.3, 8H    H3 = ,F8.3,/)                 RCS7 034
C                                                                       RCS7 035
      C      = 11.80285078                                              RCS7 036
      PI     = 3.14159265358979                                         RCS7 037
```

2

L-96a

L-96b

```
      SPI = SQRT(PI)                                                    RCS7 038
      DTR      PI / 180.0                                               RCS7 039
      RTD = 180.0/PI                                                    RCS7 040
      WC     = 2.0 * PI * FC                                            RCS7 041
      XKOO = WC/C                                                       RCS7 042
      X2KO = XKOO+XKOO                                                  RCS7 043
      X2KOA1 = X2KO*A1                                                  RCS7 044
      X2KOA2 = X2KO*A2                                                  RCS7 045
      X2KOA4 = X2KO*A4                                                  RCS7 046
C                                                                       RCS7 047
      THETA  = ASPECT * DTR                                             RCS7 048
      STHT   = SIN(THETA)                                               RCS7 049
      CTHT   = COS(THETA)                                               RCS7 050
      TANATT = STHT / CTHT                                              RCS7 051
C                                                                       RCS7 052
      SHADOW = (A4 - A2) / (H2 + H2 + H3)                               RCS7 053
      SHADOW = ATAN(SHADOW)                                             RCS7 054
      ALPHA1 = ATAN((A2-A1)/H1)                                         RCS7 055
      ALPHA2 = ATAN((A4-A2)/H3)                                         RCS7 056
      X1D = ALPHA1*RTD                                                  RCS7 057
      X2D = ALPHA2*RTD                                                  RCS7 058
      X3D = SHADOW*RTD                                                  RCS7 059
      WRITE (6,2010) ASPECT,X1D,X2D,X3D                                 RCS7 060
 2010 FORMAT ( 18H0  ASPECT ANGLE = , F8.3,/,                           RCS7 061
     1        11H  ALPHA1 = ,F8.3,/, 11H  ALPHA2 = ,F8.3,/,             RCS7 062
     2        10H  SHADOW(ALPHA3) = ,F8.3,)                             RCS7 063
```

```
  1038          1         11H  ALPHA1 = ,F8.3,/, 11H  ALPHA2 = ,F8.3,/,          RCS7 062
  1039          2         19H  SHADOW(ALPHA3) = ,F8.3 )                          RCS7 063
  1040        C                                                                  RCS7 064
  1041              SA1 = SIN(ALPHA1)                                            RCS7 065
  1042              SA2 = SIN(ALPHA2)                                            RCS7 066
  1043              A1PT = ALPHA1+THETA                                          RCS7 067

04/26/76     INPUT LISTING                         AUTOFLOW CHART SET - FWD/SCL  RADSIM

 CARD NO      ****                          CONTENTS                                 ****

  1044              A1MT = ALPHA1-THETA                                          RCS7 068
  1045              A2PT = ALPHA2+THETA                                          RCS7 069
  1046              A2MT = ALPHA2-THETA                                          RCS7 070
  1047              TANAP1 = TAN(A1PT)                                           RCS7 071
  1048              TANAP2 = TAN(A2PT)                                           RCS7 072
  1049              CA1PTS = 2.*COS(A1PT)/SA1                                    RCS7 073
  1050              CA2PTS = 2.*COS(A2PT)/SA2                                    RCS7 074
  1051        C                                                                  RCS7 075
  1052              XN1    = 1.5 - (ALPHA1 / PI)                                 RCS7 076
  1053              XN2    = 1.0 + (ALPHA1 / PI)                                 RCS7 077
  1054              XN3    = 1.0 - (ALPHA2 / PI)                                 RCS7 078
  1055              XN4    = 1.5 + (ALPHA2 / PI)                                 RCS7 079
  1056              CPON1 = COS(PI/XN1)                                          RCS7 080
  1057              CPON2 = COS(PI/XN2)                                          RCS7 081
```

3

```
C                                                                       RCS7 075
      XN1    = 1.5 - (ALPHA1 / PI)                                      RCS7 076
      XN2    = 1.0 + (ALPHA1 / PI)                                      RCS7 077
      XN3    = 1.0 - (ALPHA2 / PI)                                      RCS7 078
      XN4    = 1.5 + (ALPHA2 / PI)                                      RCS7 079
      CPON1 = COS(PI/XN1)                                               RCS7 080
      CPON2 = COS(PI/XN2)                                               RCS7 081
      CPON3 = COS(PI/XN3)                                               RCS7 082
      CPON4 = COS(PI/XN4)                                               RCS7 083
      TERMO1 = (SIN(PI / XN1)) / XN1                                    RCS7 084
      TERMO2 = (SIN(PI / XN2)) / XN2                                    RCS7 085
      TERMO3 = (SIN(PI / XN3)) / XN3                                    RCS7 086
      TERMO4 = (SIN(PI / XN4)) / XN4                                    RCS7 087
C     COEFNX ARE C(NX) TERMS                                            RCS7 088
      COEFN1= TERMO1/(CPON1 - 1.)                                       RCS7 089
      COEFN2= TERMO2/(CPON2 - 1.)                                       RCS7 090
      COEFN3= TERMO3/(CPON3 - 1.)                                       RCS7 091
      COEFN4= TERMO4/(CPON4 - 1.)                                       RCS7 092
C                                                                       RCS7 093
      IF (ASPECT .GT. 90.0) GO TO 10                                    RCS7 094
C                                                                       RCS7 095
```

```
C     DIFFRACTION TERMS ( C(N)-/+F(N,PHI) = COEFXX TERMS                RCS7 096
C          COMPUTED HERE FOR THETA.LT. 90                               RCS7 097
      COEF11 = 1.0 / ( CPON1 - COS((THETA+THETA+PI)/XN1))               RCS7 098
      COEF12 = 1.0 / ( CPON2 - COS((A1PT + A1PT)    /XN2))              RCS7 099
      COEF13 = 1.0 / ( CPON3 - COS((THETA + THETA) /XN3))               RCS7 100
      COEF14 = 1.0 / ( CPON4 - COS((A2PT + A2PT)    /XN4))              RCS7 101
      COEF21 = 1.0 / ( CPON1 - COS((THETA+THETA-PI)/XN1))               RCS7 102
      COEF22 = 1.0 / ( CPON2 - COS((A1MT + A1MT)    /XN2))              RCS7 103
      COEF24 = 1.0 / ( CPON4 - COS((A2MT + A2MT)    /XN4))              RCS7 104
C                                                                       RCS7 105
      COEF1V = COEFN1 - COEF11 * TERM01                                 RCS7 106
      COEF1H = COEFN1 + COEF11 * TERM01                                 RCS7 107
      COEF2V = COEFN2 - COEF12 * TERM02                                 RCS7 108
      COEF2H = COEFN2 + COEF12 * TERM02                                 RCS7 109
      COEF3V = COEFN3 - COEF13 * TERM03                                 RCS7 110
      COEF3H = COEFN3 + COEF13 * TERM03                                 RCS7 111
      COEF4V = COEFN4 - COEF14 * TERM04                                 RCS7 112
      COEF4H = COEFN4 + COEF14 * TERM04                                 RCS7 113
      COEF5V = COEFN1 - COEF21 * TERM01                                 RCS7 114
      COEF5H = COEFN1 + COEF21 * TERM01                                 RCS7 115
      COEF6V = COEFN2 - COEF22 * TERM02                                 RCS7 116
      COEF6H = COEFN2 + COEF22 * TERM02                                 RCS7 117
      COEF7V = COEFN4 - COEF24 * TERM04                                 RCS7 118
      COEF7H = COEFN4 + COEF24 * TERM04                                 RCS7 119
C                                                                       RCS7 120
```

```
      COEF7V = COEFN4 - COEF24 * TERM04                                RCS7 118
      COEF7H = COEFN4 + COEF24 * TERM04                                RCS7 119
C                                                                      RCS7 120
      TANAM1 = TAN(A1MT)                                               RCS7 121
      TANAM2 = TAN(A2MT)                                               RCS7 122
      CA1MTS = 2.*COS(A1MT)/SA1                                        RCS7 123
      CA2MTS = 2.*COS(A2MT)/SA2                                        RCS7 124
      Q6=C(X2KOA2 * A1MT                   )                           RCS7 125
      Q8=C(X2KOA4 * (SHADOW - THETA)       )                           RCS7 126
      GO TO 20                                                         RCS7 127
C                                                                      RCS7 128
C     DIFFRACTION TERMS ( C(N)-/+B(N,PHI) = COEFXX TERMS               RCS7 129
C        COMPUTED HERE FOR THETA.GT. 90                                RCS7 130
   10 COEF31   = 1.0 / (CPON1 - COS(2.0*(PI-A1PT)/XN1))                RCS7 131
      COEF32   = 1.0 / (CPON2 - COS(2.0*(PI-THETA)/XN2))               RCS7 132
      COEF33   = 1.0 / (CPON3 - COS(2.0*(PI-A2PT)/XN3))                RCS7 133
      COEF34   = 1.0 / (CPON4 - COS((2.0*THETA - 3.0*PI)/XN4))         RCS7 134
      COEF44   = 1.0 / (CPON4 - COS((2.0*THETA -   PI   )/XN4))        RCS7 135
                                                                       RCS7 136
      COEF1V = COEFN1 - COEF31 * TERM01                                RCS7 137
      COEF1H = COEFN1 + COEF31 * TERM01                                RCS7 138
      COEF2V = COEFN2 - COEF32 * TERM02                                RCS7 139
      COEF2H = COEFN2 + COEF32 * TERM02                                RCS7 140
      COEF3V = COEFN3 - COEF33 * TERM03                                RCS7 141
      COEF3H = COEFN3 + COEF33 * TERM03                                RCS7 142
      COEF4V = COEFN4 - COEF34 * TERM04                                RCS7 143
      COEF4H = COEFN4 + COEF34 * TERM04                                RCS7 144
      COEF5V = COEFN4 - COEF44 * TERM04                                RCS7 145
      COEF6H = COEFN4 + COEF44 * TERM04
```

```
C                                                                      RCS7 128
C     DIFFRACTION TERMS ( C(N)-/+F(N,PHI) = COEFXX TERMS               RCS7 129
C        COMPUTED HERE FOR THETA.GT. 90                                RCS7 130
   10 COEF31   = 1.0 / (CPON1 - COS(2.0*(PI-A1PT)/XN1))                RCS7 131
      COEF32   = 1.0 / (CPON2 - COS(2.0*(PI-THETA)/XN2))               RCS7 132
      COEF33   = 1.0 / (CPON3 - COS(2.0*(PI-A2PT)/XN3))                RCS7 133
      COEF34   = 1.0 / (CPON4 - COS((2.0*THETA - 3.0*PI)/XN4))         RCS7 134
      COEF44   = 1.0 / (CPON4 - COS((2.0*THETA -   PI  )/XN4))         RCS7 135
                                                                       RCS7 136
      COEF1V = COEFN1 - COEF31 * TERM01                                RCS7 137
      COEF1H = COEFN1 + COEF31 * TERM01                                RCS7 138
      COEF2V = COEFN2 - COEF32 * TERM02                                RCS7 139
      COEF2H = COEFN2 + COEF32 * TERM02                                RCS7 140
      COEF3V = COEFN3 - COEF33 * TERM03                                RCS7 141
      COEF3H = COEFN3 + COEF33 * TERM03                                RCS7 142
      COEF4V = COEFN4 - COEF34 * TERM04                                RCS7 143
      COEF4H = COEFN4 + COEF34 * TERM04                                RCS7 144
      COEF5V = COEFN4 - COEF44 * TERM04                                RCS7 145
      COEF5H = COEFN4 + COEF44 * TERM04                                RCS7 146
C                                                                      RCS7 147
      Q1=Q(X2KOA1 * (PI-A1PT)                 )                        RCS7 148
      Q2=Q(X2KOA2 * (PI - SHADOW - THETA))                             RCS7 149
      Q3=Q(X2KOA2 * (PI-A2PT)                 )                        RCS7 150
   20 CONTINUE                                                         RCS7 151
C                                                                      RCS7 152
      DO 900 I = NMIN, NMAX                                            RCS7 153
```

3

L-96C

L-96d

```
      XI      = I - 1                                            RCS7 154
      W       = (2. * PI * XI * DF) / 1000.0                     RCS7 155
      XKO     = W / C                                            RCS7 156
      XKO2C     = XKO * (CTHT + CTHT)                            RCS7 157
C                                                                RCS7 158
      TAU1        = XKO * A1 * CA1PTS                            RCS7 159
      TAU2        = XKO * A2 * CA1PTS                            RCS7 160
      TAU3        = XKO * A3 * CA2PTS                            RCS7 161
      TAU4        = XKO * A4 * CA2PTS                            RCS7 162
      FTAU1   = F       (TAU1)                                   RCS7 163
      FTAU2   = F       (TAU2)                                   RCS7 164
      FTAU3   = F       (TAU3)                                   RCS7 165
      FTAU4   = F       (TAU4)                                   RCS7 166
C                                                                RCS7 167
      SI1     = 2.0 * XKO * A1 * STHT                            RCS7 168
      SI2     = 2.0 * XKO * A2 * STHT                            RCS7 169
      SI4     = 2.0 * XKO * A4 * STHT                            RCS7 170
      CALL BESL  (SI1, XJOX1, XJ1X1, XJ2X1)                      RCS7 171
      CALL BESL  (SI2, XJOX2, XJ1X2, XJ2X2)                      RCS7 172
      CALL BESL  (SI4, XJOX4, XJ1X4, XJ2X4)                      RCS7 173
      TERM1P = CMPLX(XJOX1,XJ1X1)                                RCS7 174
      TERM1M = CONJG(TERM1P)                                     RCS7 175
      TERM2P = CMPLX(XJOX2,XJ1X2)                                RCS7 176
      TERM2M = CONJG(TERM2P)                                     RCS7 177
      TERM4P = CMPLX(XJOX4,XJ1X4)                                RCS7 178
```

```
  1153             TERM2M = CONJG(TERM2P)
  1154             TERM4P = CMPLX(XJ0X4,XJ1X4)                                       RCS7 178
  1155             TERM4M = CONJG(TERM4P)                                            RCS7 179
  1156             TERM5  = XJ0X1 + XJ2X1                                            RCS7 180
  1157             TERM6  = XJ0X2 + XJ2X2                                            RCS7 181
  1158             TERM7  = XJ0X4 + XJ2X4                                            RCS7 182
  1159       C                                                                       RCS7 183
04/26/76     INPUT LISTING                          AUTOFLOW CHART SET - FWD/SCL  RADSIM

 CARD NO     ****                            CONTENTS                                    ****

  1160             PHI1    = XK02C * (H1+H2)                                         RCS7 184
  1161             PHI2    = XK02C * (H2)                                            RCS7 185
  1162             PHI3    = PHI2                                                    RCS7 186
  1163             PHI4    = XK02C * (H2+H3)                                         RCS7 187
  1164             PHASE1 = CMPLX(COS(PHI1),-SIN(PHI1))                              RCS7 188
  1165             PHASE2 = CMPLX(COS(PHI2),-SIN(PHI2))                              RCS7 189
  1166             PHASE3 = CMPLX(COS(PHI3), SIN(PHI3))                              RCS7 190
  1167             PHASE4 = CMPLX(COS(PHI4), SIN(PHI4))                              RCS7 191
  1168       C                                                                       RCS7 192
  1169             IF (ASPECT .GT. 90.0) GO TO 30                                    RCS7 193
  1170       C                                                                       RCS7 194
  1171             TAU5    = XK0 * A1 * CA1MTS                                       RCS7 195
  1172             TAU6    = XK0 * A2 * CA1MTS                                       RCS7 196
  1173             TAU7    = XK0 * A2 * CA2MTS                                       RCS7 197
  1174             TAU8    = XK0 * A4 * CA2MTS                                       RCS7 198
  1175             FTAU5  = F      (TAU5)                                            RCS7 199
  1176             FTAU6  = F      (TAU6)                                            RCS7 200
  1177             FTAU7  = F      (TAU7)                                            RCS7 201
```

```
      PHASE1 = CMPLX(COS(PHI1),-SIN(PHI1))                          RCS7 188
      PHASE2 = CMPLX(COS(PHI2),-SIN(PHI2))                          RCS7 189
      PHASE3 = CMPLX(COS(PHI3), SIN(PHI3))                          RCS7 190
      PHASE4 = CMPLX(COS(PHI4), SIN(PHI4))                          RCS7 191
C                                                                   RCS7 192
      IF (ASPECT .GT. 90.0) GO TO 30                                RCS7 193
C                                                                   RCS7 194
      TAU5     = XKO * A1 * CA1MTS                                  RCS7 195
      TAU6     = XKO * A2 * CA1MTS                                  RCS7 196
      TAU7     = XKO * A2 * CA2MTS                                  RCS7 197
      TAU8     = XKO * A4 * CA2MTS                                  RCS7 198
      FTAU5  = F     (TAU5)                                         RCS7 199
      FTAU6  = F     (TAU6)                                         RCS7 200
      FTAU7  = F     (TAU7)                                         RCS7 201
      FTAU8  = F     (TAU8)                                         RCS7 202
C                                                                   RCS7 203
      FFVV01 = A1 * TERM1M * COEF1V * PHASE1                        RCS7 204
      FFHH01 = A1 * TERM1M * COEF1H * PHASE1                        RCS7 205
      FFVV02 = A2 * TERM2M * COEF2V * PHASE2                        RCS7 206
      FFHH02 = A2 * TERM2M * COEF2H * PHASE2                        RCS7 207
      FFVV03 = A2 * TERM2M * COEF3V * PHASE3                        RCS7 208
      FFHH03 = A2 * TERM2M * COEF3H * PHASE3                        RCS7 209
      FFVV04 = A4 * TERM4M * COEF4V * PHASE4                        RCS7 210
      FFHH04 = A4 * TERM4M * COEF4H * PHASE4                        RCS7 211
```

3

```
      FFVV05 = A1 * TERM1P * COEFF5V * PHASE1                     RCS7 212
      FFHH05 = A1 * TERM1P * COEFF5H * PHASE1                     RCS7 213
      FFVV06 = A2 * TERM2P * COEF6V * PHASE2 * Q6                 RCS7 214
      FFHH06 = A2 * TERM2P * COEF6H * PHASE2 * Q6                 RCS7 215
      FFVV07 = A4 * TERM4P * COEF7V * PHASE4 * Q8                 RCS7 216
      FFHH07 = A4 * TERM4P * COEF7H * PHASE4 * Q8                 RCS7 217
      FFVV08 = A1 * TERM1M * TANAP1 * ( 0.5) * FTAU1 * PHASE1     RCS7 218
      FFHH08 = -FFVV08                                            RCS7 219
      FFVV09 = A2 * TERM2M * TANAP1 * (-0.5) * FTAU2 * PHASE2     RCS7 220
      FFHH09 = -FFVV09                                            RCS7 221
      FFVV10 = A2 * TERM2M * TANAP2 * ( 0.5) * FTAU3 * PHASE3     RCS7 222
      FFHH10 = -FFVV10                                            RCS7 223
      FFVV11 = A4 * TERM4M * TANAP2 * (-0.5) * FTAU4 * PHASE4     RCS7 224
      FFHH11 = -FFVV11                                            RCS7 225
      FFVV12 = A1 * TERM1P * TANAM1 * ( 0.5) * FTAU5 * PHASE1 * Q6 RCS7 226
      FFHH12 = -FFVV12                                            RCS7 227
      FFVV13 = A2 * TERM2P * TANAM1 * (-0.5) * FTAU6 * PHASE2 * Q6 RCS7 228
      FFHH13 = -FFVV13                                            RCS7 229
      FFVV14 = A2 * TERM2P * TANAM2 *(-.5)*(1.-FTAU7)* PHASE3 * Q8 RCS7 230
      FFHH14 = -FFVV14                                            RCS7 231
      FFVV15 = A4 * TERM4P * TANAM2 * (-0.5) * FTAU8 * PHASE4 * Q8 RCS7 232
      FFHH15 = -FFVV15                                            RCS7 233
      FFVV16 = -A1 * COEFN1 * TERM5  * PHASE1                     RCS7 234
      FFHH16 =  FFVV16                                            RCS7 235
      FFVV17 = -A2 * COEFN2 * TERM6  * PHASE2                     RCS7 236
      FFHH17 =  FFVV17                                            RCS7 237
```

```
  1210          FFVV16 = -A1 * COEFN1 * TERM5  * PHASE1                     RCS7 234
  1211          FFHH16 =  FFVV16                                            RCS7 235
  1212          FFVV17 = -A2 * COEFN2 * TERM6  * PHASE2                     RCS7 236
  1213          FFHH17 =  FFVV17                                            RCS7 237
  1214          FFVV18 = -A2 * COEFN3 * TERM6  * PHASE3                     RCS7 238
  1215          FFHH18 =  FFVV18                                            RCS7 239
  1216          FFVV19 = -A4 * COEFN4 * TERM7  * PHASE4                     RCS7 240
  1217          FFHH19 =  FFVV19                                            RCS7 241




04/26/76     INPUT LISTING                   AUTOFLOW CHART SET - FWD/SCL  RADSIM

 CARD NO       ****                         CONTENTS                            ****

  1218          FFVV20 = -A1 * COEFN1 * TERM5  * PHASE1                     RCS7 242
  1219          FFHH20 =  FFVV20                                            RCS7 243
  1220          FFVV21 = -A2 * COEFN2 * TERM6  * PHASE2 * Q6                RCS7 244
  1221          FFHH21 =  FFVV21                                            RCS7 245
  1222          FFVV22 = -A4 * COEFN4 * TERM7  * PHASE4 * Q8                RCS7 246
  1223          FFHH22 =  FFVV22                                            RCS7 247
  1224       C                                                              RCS7 248
  1225          FFVV   = FFVV01 + FFVV02 + FFVV03 + FFVV04 + FFVV05 + FFVV06 +   RCS7 249
  1226         1         FFVV07 + FFVV08 + FFVV09 + FFVV10 + FFVV11 + FFVV12 +   RCS7 250
  1227         2         FFVV13 + FFVV14 + FFVV15 + FFVV16 + FFVV17 + FFVV18 +   RCS7 251
  1228         3         FFVV19 + FFVV20 + FFVV21 + FFVV22                      RCS7 252
  1229          FFHH   = FFHH01 + FFHH02 + FFHH03 + FFHH04 + FFHH05 + FFHH06 +   RCS7 253
```

```
                                                                      RCS7 243
      FFVV21 = -A2 * COEFN2 * TERM6  * PHASE2 * Q6                    RCS7 244
      FFHH21 =  FFVV21                                                RCS7 245
      FFVV22 = -A4 * COEFN4 * TERM7  * PHASE4 * Q8                    RCS7 246
      FFHH22 =  FFVV22                                                RCS7 247
C                                                                     RCS7 248
      FFVV   = FFVV01 + FFVV02 + FFVV03 + FFVV04 + FFVV05 + FFVV06 +  RCS7 249
     1         FFVV07 + FFVV08 + FFVV09 + FFVV10 + FFVV11 + FFVV12 +  RCS7 250
     2         FFVV13 + FFVV14 + FFVV15 + FFVV16 + FFVV17 + FFVV18 +  RCS7 251
     3         FFVV19 + FFVV20 + FFVV21 + FFVV22                      RCS7 252
      FFHH   = FFHH01 + FFHH02 + FFHH03 + FFHH04 + FFHH05 + FFHH06 +  RCS7 253
     1         FFHH07 + FFHH08 + FFHH09 + FFHH10 + FFHH11 + FFHH12 +  RCS7 254
     2         FFHH13 + FFHH14 + FFHH15 + FFHH16 + FFHH17 + FFHH18 +  RCS7 255
     3         FFHH19 + FFHH20 + FFHH21 + FFHH22                      RCS7 256
C                                                                     RCS7 257
      GO TO 40                                                        RCS7 258
C                                                                     RCS7 259
   30 CONTINUE                                                        RCS7 260
      FFVV01 = A1 * TERM1M * COEF1V * PHASE1 * Q1                     RCS7 261
      FFHH01 = A1 * TERM1M * COEF1H * PHASE1 * Q1                     RCS7 262
      FFVV02 = A2 * TERM2M * COEF2V * PHASE2 * Q2                     RCS7 263
      FFHH02 = A2 * TERM2M * COEF2H * PHASE2 * Q2                     RCS7 264
      FFVV03 = A2 * TERM2M * COEF3V * PHASE3 * Q3                     RCS7 265
      FFHH03 = A2 * TERM2M * COEF3H * PHASE3 * Q3                     RCS7 266
      FFVV04 = A4 * TERM4M * COEF4V * PHASE4                          RCS7 267
      FFHH04 = A4 * TERM4M * COEF4H * PHASE4                          RCS7 268
      FFVV05 = A4 * TERM4P * COEF5V * PHASE4                          RCS7 269
```

3

L-96e

L-96f

```
      FFHH05 = A4 * TERM4P * COEF5H * PHASE4                                RCS7 270
      FFVV06 = A1 * TERM1M * TANAP1 * ( 0.5) * FTAU1 * PHASE1 * Q1          RCS7 271
      FFHH06 = -FFVV06                                                      RCS7 272
      FFVV07 = A2 * TERM2M * TANAP1 * (-0.5) * FTAU2 * PHASE2 * Q1          RCS7 273
      FFHH07 = -FFVV07                                                      RCS7 274
      FFVV08 = A2 * TERM2M * TANAP2 * ( 0.5) * FTAU3 * PHASE3 * Q3          RCS7 275
      FFHH08 = -FFVV08                                                      RCS7 276
      FFVV09 = A2 * TERM2M * TANAT1 * ( 0.5)          * PHASE3 *(Q2-Q3)     RCS7 277
      FFHH09 = -FFVV09                                                      RCS7 278
      FFVV11 = A4 * TERM4M * TANAP2 * (-0.5) * FTAU4 * PHASE4 * Q3          RCS7 279
      FFHH11 = -FFVV11                                                      RCS7 280
      FFVV12 = -A1 * COEFN1 * TERM5  * PHASE1 * Q1                          RCS7 281
      FFHH12 =  FFVV12                                                      RCS7 282
      FFVV13 = -A2 * COEFN2 * TERM6  * PHASE2 * Q2                          RCS7 283
      FFHH13 =  FFVV13                                                      RCS7 284
      FFVV14 = -A2 * COEFN3 * TERM6  * PHASE3 * Q3                          RCS7 285
      FFHH14 =  FFVV14                                                      RCS7 286
      FFVV15 = -A4 * COEFN4 * TERM7  * PHASE4 *Q8                           RCS7 287
      FFHH15 =  FFVV15                                                      RCS7 288
      FFVV16 = -A4 * COEFN4 * TERM7  * PHASE4                               RCS7 289
      FFHH16 =  FFVV16                                                      RCS7 290
C                                                                           RCS7 291
      FFVV   = FFVV01 + FFVV02 + FFVV03 + FFVV04 + FFVV05 + FFVV06 +        RCS7 292
     1         FFVV07 + FFVV08 + FFVV09          + FFVV11 + FFVV12 +        RCS7 293
```

```
                                                                                RCS7 291
1268          FFVV   = FFVV01 + FFVV02 + FFVV03 + FFVV04 + FFVV05 + FFVV06 +    RCS7 292
1269        1          FFVV07 + FFVV08 + FFVV09          + FFVV11 + FFVV12 +    RCS7 293
1270        2          FFVV13 + FFVV14 + FFVV15 + FFVV16                        RCS7 294
1271     C                                                                      RCS7 295
1272          FFHH   = FFHH01 + FFHH02 + FFHH03 + FFHH04 + FFHH05 + FFHH06 +    RCS7 296
1273        1          FFHH07 + FFHH08 + FFHH09          + FFHH11 + FFHH12 +    RCS7 297
1274        2          FFHH13 + FFHH14 + FFHH15 + FFHH16                        RCS7 298
1275     C                                                                      RCS7 299




04/26/76      INPUT LISTING                     AUTOFLOW CHART SET - FWD/SCL  RADSIM

 CARD NO      ****                      CONTENTS                                  ****

 1276     C                                                                     RCS7 300
 1277        40 FFVV   = FFVV * .02539998 * SPI                                 RCS7 301
 1278           FFHH   = FFHH * .02539998 * SPI                                 RCS7 302
 1279     C                                                                     RCS7 303
 1280           CFVV   =-CONJG(FFVV)                                            RCS7 304
 1281           CFHH   = CONJG(FFHH)                                            RCS7 305
 1282     C                                                                     RCS7 306
 1283           EVVR(I) =REAL(CFVV)                                             RCS7 307
 1284           EVVI(I) =AIMAG(CFVV)                                            RCS7 308
 1285           EHHR(I) =REAL(CFHH)                                             RCS7 309
 1286           EHHI(I) =AIMAG(CFHH)                                            RCS7 310
```

```
      FFHH   = FFHH * .02539998 * SPI                                   RCS7 302
C                                                                       RCS7 303
      CFVV    =-CONJG(FFVV)                                             RCS7 304
      CFHH    = CONJG(FFHH)                                             RCS7 305
C                                                                       RCS7 306
      EVVR(I) =REAL(CFVV)                                               RCS7 307
      EVVI(I) =AIMAG(CFVV)                                              RCS7 308
      EHHR(I) =REAL(CFHH)                                               RCS7 309
      EHHI(I) =AIMAG(CFHH)                                              RCS7 310
C                                                                       RCS7 311
  900 CONTINUE                                                          RCS7 312
C                                                                       RCS7 313
      RETURN                                                            RCS7 314
      END                                                               RCS7 315
```

3

```
      SUBROUTINE BESL ( X, BO, B1, B2 )                                 RCS7 316
C                                                                       RCS7 317
C    * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS   RCS7 318
C    * COMPUTES JO,J1,OR J2 FOR POSITIVE REAL ARGUMENTS                 RCS7 319
C    * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )RCS7 320
C                                                                       RCS7 321
      S = 1.0                                                           RCS7 322
      IF (X .LT. 0.0) S=-1.0                                            RCS7 323
      X = ABS(X)                                                        RCS7 324
C                                                                       RCS7 325
      IF ( X .GT. 1.E-6 ) GO TO 5                                       RCS7 326
      BO = 1.0                                                          RCS7 327
      B1 = 0.0                                                          RCS7 328
      B2 = 0.0                                                          RCS7 329
      X = X * S                                                         RCS7 330
      RETURN                                                            RCS7 331
C                                                                       RCS7 332
    5 CONTINUE                                                          RCS7 333
C                                                                       RCS7 334
    1 IF ( X .GE. 3.) GO TO 9                                           RCS7 335
      X1 = X/3.                                                         RCS7 336
      X1 = X1*X1                                                        RCS7 337
      BO = 1.+ X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479RCS7 338
     1 +     X1*(-.0039444+ X1*2.1E-4 )))) )                            RCS7 339
```

2

```
1315            1 +       X1*(-.0039444+ X1*2.1E-4 )))) )                      RCS7 339
1316              GO TO 10                                                     RCS7 340
1317        C                                                                  RCS7 341
1318            9 X2 = 3./X                                                    RCS7 342
1319              F0 =  .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*  RCS7 343
1320            1       (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 ))))) )     RCS7.344
1321              T0 = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573  RCS7 345
1322            1     +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))    RCS7 346
1323              B = F0*COS(T0)/SQRT(X)                                       RCS7 347
1324        C                                                                  RCS7 348
1325           10 BC = B                                                       RCS7 349
1326        C                                                                  RCS7 350
1327            2 IF ( X .GE. 3. ) GO TO 19                                    RCS7 351
1328              X1 = X/3.                                                    RCS7 352
1329              X1 = X1*X1                                                   RCS7 353
1330              B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*   RCS7 354
1331            1              (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )   RCS7 355
1332              GO TO 20                                                     RCS7 356
1333        C                                                                  RCS7 357

04/26/76       INPUT LISTING               AUTOFLOW CHART SET - FWD/SCL  RADSIM

CARD NO        ****                    CONTENTS                             ****

1334            19 X2 = 3./X                                                   RCS7 358
```

```
 1328             X1 = X/3.                                                      RCS7 352
 1329             X1 = X1*X1                                                     RCS7 353
 1330             B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*  RCS7 354
 1331            1           (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )     RCS7 355
 1332             GO TO 20                                                       RCS7 356
 1333       C                                                                    RCS7 357
04/26/76     INPUT LISTING              AUTOFLOW CHART SET - FWC/SCL  RADSIM
  CARD NO       ****                      CONTENTS                                  ****
 1334          19 X2 = 3./X                                                      RCS7 358
 1335             F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*  RCS7 359
 1336            1    (-.00249511 +X2*(.00113653 -.00020033*X2 )))))             RCS7 360
 1337             T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879 RCS7 361
 1338            1    +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))         RCS7 362
 1339             B = F1*COS(T1)/SQRT(X)                                         RCS7 363
 1340       C                                                                    RCS7 364
 1341          20 B1 = B * S                                                     RCS7 365
 1342             X = X * S                                                      RCS7 366
 1343             B2= (2./X)*B1 - B0                                             RCS7 367
 1344          50 RETURN                                                         RCS7 368
 1345             END                                                            RCS7 369
```

3

L-969

496-7

```
      COMPLEX FUNCTION F(TAU)                                            RCS7 370
C                                                                        RCS7 371
C     COMPUTES FTAU WHERE FTAU =(EXP(-J*TAU**2)/2*TAU)*SQRT(PI/2.)*      RCS7 372
C                                (C2(TAU**2) + J*S2(TAU**2))             RCS7 373
C                                                                        RCS7 374
      COMPLEX B, FP                                                      RCS7 375
      PI = 3.14159265358979                                              RCS7 376
      PIO2 = PI/2.                                                       RCS7 377
      C1 = SQRT(PI/2.)                                                   RCS7 378
      C2 = 1./C1                                                         RCS7 379
      ATAUS = ABS(TAU)                                                   RCS7 380
      IF (ATAUS .LE. 0.5  )GO TO 20                                      RCS7 381
C                                                                        RCS7 382
C     FOR TAUS .GT. 0.5, FUNCTION COMPUTED USING POLYNOMIAL              RCS7 383
C                        APPROXIMATION                                   RCS7 384
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,             RCS7 385
C   *         SECTIONS  7.3.9,7.3.10,7.3.32,7.3.33)                      RCS7 386
      TAUS = SQRT(ATAUS)                                                 RCS7 387
      X = C2*TAUS                                                        RCS7 388
      XS = X*X                                                           RCS7 389
C                                                                        RCS7 390
      FX = (1.0+0.926*X')/(2.0+1.792*X+3.104*XS)                         RCS7 391
      GX = 1.0/(2.0+4.142*X+3.492*XS+6.67*X*XS)                          RCS7 392
C                                                                        RCS7 393
      CC1XS = COS(ATAUS)                                                 RCS7 394
      SC1XS = SIN(ATAUS)                                                 RCS7 395
```

```
C                                                                        RCS7 393
      CCIXS = COS(ATAUS)                                                 RCS7 394
      SCIXS = SIN(ATAUS)                                                 RCS7 395
C                                                                        RCS7 396
      CX = 0.5 + FX*SCIXS - GX*CCIXS                                     RCS7 397
      SX = 0.5 - FX*CCIXS - GX*SCIXS                                     RCS7 398
C                                                                        RCS7 399
      IF (TAU .LT. 0.0) GO TO 10                                         RCS7 400
      B = CMPLX(CX,SX)                                                   RCS7 401
      FP = CMPLX(  COS(ATAUS), -SIN(ATAUS) )                             RCS7 402
      F = (C1*B*FP)/TAUS                                                 RCS7 403
      RETURN                                                             RCS7 404
C                                                                        RCS7 405
   10 CONTINUE                                                           RCS7 406
      B = CMPLX(SX,CX)                                                   RCS7 407
      A = ATAUS-PIO2                                                     RCS7 408
      FP = CMPLX( COS(A),SIN(A) )                                        RCS7 409
      F = (B*FP*C1)/TAUS                                                 RCS7 410
      RETURN                                                             RCS7 411
C                                                                        RCS7 412
   20 CONTINUE                                                           RCS7 413
C     FOR TAUS .LE. 0.5, FUNCTION IS EXPANDED IN SERIES AND FIRST FEW    RCS7 414
C                             TERMS INTEGRATED TERM BY TERM TO OBTAIN RESULT RCS7 415
```

```
1384          A = ATAUS-PI02                                              RCS7 408
1385          FP = CMPLX( COS(A),SIN(A) )                                 RCS7 409
1386          F = (B*FP*C1)/TAUS                                          RCS7 410
1387          RETURN                                                      RCS7 411
1388    C                                                                 RCS7 412
1389       20 CONTINUE                                                    RCS7 413
1390    C     FOR TAUS .LE. 0.5, FUNCTION IS EXPANDED IN SERIES AND FIRST FEW    RCS7 414
1391    C                   TERMS INTEGRATED TERM BY TERM TO OBTAIN RESULT RCS7 415
```

```
04/26/76        INPUT LISTING                AUTOFLOW CHART SET - FWD/SCL  RADSIM

3 CARD NO       ****                      CONTENTS                         ****

1392          FP = CMPLX(COS(TAU),-SIN(TAU))                              RCS7 416
1393          TS = TAU*TAU                                                RCS7 417
1394          FR = 1 - TS*( .1 - .0046296296*TS)                          RCS7 418
1395          FI = TAU *( .333333333 - TS*(.0238095238 - 7.57575757E-4*TS))  RCS7 419
1396          B = CMPLX(FR,FI)                                            RCS7 420
1397          F = FP*B                                                    RCS7 421
1398          RETURN                                                      RCS7 422
1399          END                                                         RCS7 423
```

```
      FUNCTION Q(Z)                                                    RCS7 424
C     Q(Z) = 0.5*(1 + ERF(Z))                                          RCS7 425
C   * ERF(Z) IS EVALUATED USING A RATIONAL POLYNOMIAL APPROXIMATION    RCS7 426
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,           RCS7 427
C   *                SECTION 7.1.26)                                   RCS7 428
C                                                                      RCS7 429
      IF ( Z.GT. 2.) GO TO 10                                          RCS7 430
      IF ( Z.LT.-2.) GO TO 20                                          RCS7 431
      AZ = ABS(Z)                                                      RCS7 432
      P = 1.0/(1.0 + .47047*AZ)                                        RCS7 433
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)   RCS7 434
      IF (Z) 2,4,6                                                     RCS7 435
    2 Q = (1.0 - Y)/2.                                                 RCS7 436
      RETURN                                                           RCS7 437
    4 Q = .5                                                           RCS7 438
      RETURN                                                           RCS7 439
    6 Q = (1.0 + Y)/2.                                                 RCS7 440
      RETURN                                                           RCS7 441
   10 Q = 1.                                                           RCS7 442
      RETURN                                                           RCS7 443
   20 Q = 0.                                                           RCS7 444
      RETURN                                                           RCS7 445
      END                                                              RCS7 446
```

L.6 CYLINDER-FRUSTRUM COMBINATION

The far-field scattering from a cylinder-frustrum combination target shown in Figure L.6.1 has been formulated using the Ruck-Ufimtsev technique (Ref. 7). The solution can be represented in the following form:

$$e(\theta) = \mp \sqrt{\pi} \Big\{ g(11) + g(12) + g(1) + g(2)$$
$$+ g(3) + g(4) + g(9) + g(5) + g(6)$$
$$+ g(7) + g(8) \Big\}$$

where g(m) represents the sum of the fringe wave scattering and the physical optics response associated with edge m and the upper and lower signs correspond to vertical and horizontal polarization, respectively.

With the use of the diffraction coefficients at the concave edges (F #1), the expressions for the g(m) are the following:

For $0 < \theta < \pi/2$

$$g(11) = a_1 e^{ip_{11}} \Big\{ JJ_{11+} \Big[C(n_c) \mp B(n_c, \tfrac{\pi}{2} - \theta) \Big] - C(n_c) JJ_{21} \Big\}$$

$$g(12) = a_1 e^{ip_{11}} \Big\{ JJ_{11-} \Big[C(n_c) \mp B(n_c, \tfrac{\pi}{2} + \theta) - C(n_c) JJ_{21} \Big\}$$

$$g(2) = a_2 e^{ip_2} \Big\{ JJ_{12-} \Big[C(n_2) \mp B(n_2, \theta+\alpha) \mp 0.5 \tan (\alpha+\theta) F_2 \Big]$$
$$- C(n_2) JJ_{22} \Big\}$$

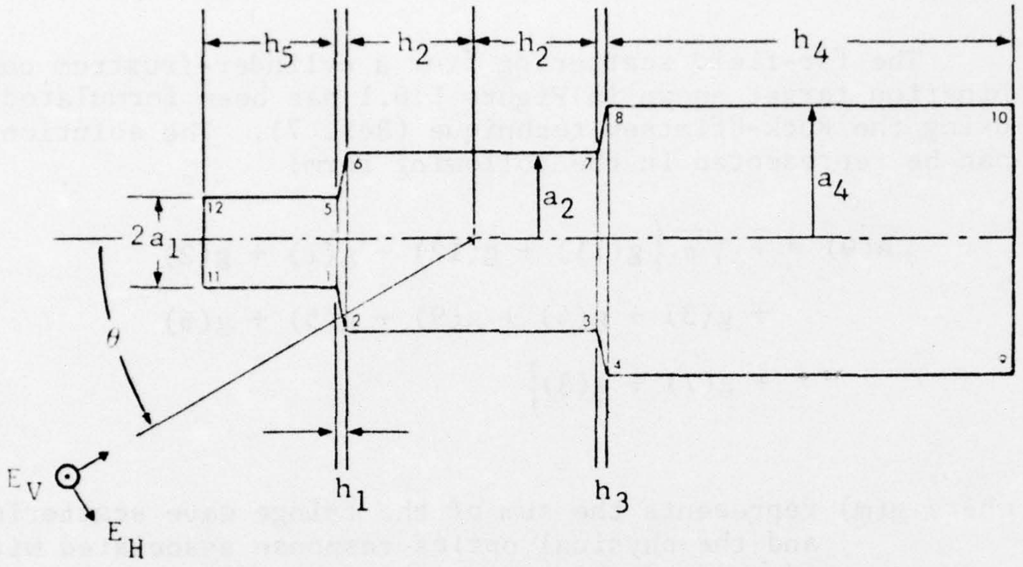


Fig. L.6-1 GEOMETRY OF CYLINDER-FRUSTRUM COMBINATION

$$\begin{aligned} g(4) &= a_4 e^{ip4} \Big\{ JJ_{14-} \big[C(n_4) \mp B(n_4,\theta+\alpha) \mp 0.5 \tan(\alpha+\theta) F_4 \big] \\ &\qquad - C(n_4) JJ_{24} \Big\} \end{aligned}$$

$$g(9) = a_4 e^{ip9} \Big\{ JJ_{14-} \big[C(n_c) \mp B(n_c,\theta) \big] - C(n_c) JJ_{24} \Big\}$$

$$\begin{aligned} g(1) &= a_1 e^{ip1} \Big\{ JJ_{11-} \big[C(n_1) \mp B(n_1,\theta) \pm 0.5 \tan(\alpha+\theta) F_1 \big] \\ &\qquad - C(n_1) JJ_{21} \Big\} \end{aligned}$$

$$\begin{aligned} g(3) &= a_3 e^{ip_3} \Big\{ JJ_{12-} \big[C(n_3) \mp B(n_3,\theta) \pm 0.5 \tan(\alpha+\theta) F_3 \big] \\ &\qquad - C(n_3) JJ_{22} \Big\} \end{aligned}$$

$$g(5) = a_1 e^{ip1}\ JJ_{11+} \Big\{ \mp 0.5 \tan(\alpha-\theta) \big[1 - F_5 \big] Q_6 \Big\}$$

$$\begin{aligned} g(6) &= a_2 e^{ip_2} \Big\{ JJ_{12+} \big[C(n_2) \mp B(n_2,\alpha-\theta) \mp 0.5 \tan(\alpha-\theta) F_6 \big] \\ &\qquad - C(n_2) JJ_{22} \Big\} Q_6 \end{aligned}$$

$$g(7) = a_2 e^{ip_3}\ JJ_{12+} \Big\{ \mp 0.5 \tan(\alpha-\theta) \big[1 - F_7 \big] \Big\} Q_8$$

$$\begin{aligned} g(8) &= a_4 e^{ip4} \Big\{ JJ_{14+} \big[C(n_4) \mp B(n_4,\alpha-\theta) \mp 0.5 \tan(\alpha-\theta) F_8 \big] \\ &\qquad - C(n_4) JJ_{24} \Big\} Q_8 \end{aligned}$$

For $\pi/2 < \theta < \pi$

$$g(5) = g(6) = g(7) = g(8) = 0$$

$$g(9) = a_4 e^{ip_9} \Big\{ JJ_{14-} \Big[C(n_c) \mp B(n_c,\theta) \Big] - C(n_c) JJ_{24} \Big\}$$

$$g(10) = a_4 e^{ip_9} \Big\{ JJ_{14+} \Big[C(n_c) \mp B(n_c,\theta - \tfrac{\pi}{2}) \Big] - C(n_c) JJ_{24} \Big\}$$

$$\begin{aligned} g(4) = a_4 e^{ip_4} \Big\{ JJ_{14-} \Big[&C(n_4) \mp B(n_4,\pi-\theta) \mp 0.5 \tan(\alpha+\theta) F_4 Q_1 \Big] \\ &- C(n_4) JJ_{24} \Big\} \end{aligned}$$

$$\begin{aligned} g(2) = a_2 e^{ip_2} \Big\{ JJ_{12-} \Big[&C(n_2) Q_2 \mp B(n_2,\pi-\theta) Q_2 \\ &\mp 0.5 \tan(\alpha+\theta) F_2 Q_1 \Big] - C(n_2) JJ_{22} Q_2 \Big\} \end{aligned}$$

$$g(11) = a_1 e^{ip_{11}} \Big\{ JJ_{11-} \Big[C(n_c) \mp B(n_c,\pi-\theta) \Big] - C(n_c) JJ_{21} \Big\} Q_{11}$$

$$\begin{aligned} g(1) = a_1 e^{ip_1} \Big\{ JJ_{11-} \Big[&C(n_1) \mp B(n_1,\pi-\alpha-\theta) \\ &\pm 0.5 \tan(\alpha+\theta) F_1 \Big] Q_1 \\ &- C(n_1) JJ_{21} Q_1 \pm 0.5\, JJ_{11-} \tan\theta\, Q_{11} \Big[1-Q_1\Big]\Big] \Big\} \end{aligned}$$

$$\begin{aligned} g(3) = a_2 e^{ip_3} \Big\{ JJ_{12-} \Big[&C(n_3) \mp B(n_3,\pi-\alpha-\theta) \\ &\pm 0.5 \tan(\alpha+\theta) F_3 \Big] Q_3 \\ &- C(n_3) JJ_{22} Q_3 \pm 0.5\, JJ_{12-} \tan\theta\, Q_2 \Big[1-Q_3\Big]\Big] \Big\} \end{aligned}$$

where the upper and lower signs in the previous expressions correspond to vertical and horizontal polarizations, respectively, and

$$C(n) = \frac{\sin\frac{\pi}{n}}{n}\left(\frac{1}{\cos\frac{\pi}{n} - 1}\right)$$

$$B(n,\psi) = \frac{\sin\frac{\pi}{n}}{n}\left(\frac{1}{\cos\frac{\pi}{n} - \cos\frac{2\psi}{n}}\right)$$

$$JJ_{1m\mp} = \left[J_o(2ka_m\sin\theta) \mp i\ J_1(2ka_m\sin\theta)\right]$$

$$JJ_{2m} = \left[J_o(2ka_m\sin\theta) + J_2(2ka_m\sin\theta)\right]$$

$$n_1 = n_3 = 1 - \frac{\alpha}{\pi}$$

$$n_2 = n_4 = 1 + \frac{\alpha}{\pi}$$

$$n_c = 3/2$$

$$p_{11} = -2k(h_1 + h_2 + h_5)\cos\theta$$

$$p_1 = -2k(h_1 + h_2)\cos\theta$$

$$p_2 = -2k\ h_2\cos\theta$$

$$p_3 = 2k\ h_2\cos\theta$$

$$p_4 = 2k(h_2 + h_3)\cos\theta$$

$$p5 = 2k(h2 + h3 + h4)\cos\theta$$

$$Q_{\begin{pmatrix}6\\8\end{pmatrix}} = Q(2ka_{\begin{pmatrix}2\\4\end{pmatrix}}(\alpha_{\begin{pmatrix}4\\3\end{pmatrix}}-\theta))$$

$$Q_{\begin{pmatrix}1\\2\\11\end{pmatrix}} = Q(2ka_{\begin{pmatrix}1\\2\\1\end{pmatrix}}\ (\pi-\alpha_{\begin{pmatrix}1\\3\\4\end{pmatrix}} - \theta))$$

$$\tau^2_{\begin{pmatrix}1\\2\\3\\4\end{pmatrix}} = 2ka_{\begin{pmatrix}1\\2\\2\\4\end{pmatrix}}\ \csc\alpha\ \cos(\alpha+\theta)$$

$$\tau^2_{\begin{pmatrix}5\\6\\7\\8\end{pmatrix}} = 2ka_{\begin{pmatrix}1\\2\\2\\4\end{pmatrix}}\ \csc\alpha\ \cos(\alpha-\theta)$$

$$\alpha = \tan^{-1}\frac{a_2-a_1}{h_1} = \tan^{-1}\frac{a_4-a_2}{h_3}$$

$$\alpha_3 = \tan^{-1}\frac{a_4-a_2}{h_2+h_2+h_3}$$

$$\alpha_4 = \tan^{-1}\frac{a_2-a_1}{h_1+h_5}$$

$$F_m = F(\tau_m) = \frac{e^{-i\tau^2{}_m}}{\tau_m}\int_o^{\tau_m} e^{it^2}dt.$$

The expression of the g(m) modified for use in the second formulation (F #2) of the scattering from the concave edges (1, 3, 5, 7) are the following:

For $\theta < \pi/2$

$$g(1) = \pm a_1 e^{ip_1} JJ_{11-} (0.5) \left[\tan(\alpha+\theta) (F_1-1) + \tan\theta \right]$$

$$g(3) = \pm a_3 e^{ip_3} JJ_{12-} (0.5) \left[\tan(\alpha+\theta) (F_3-1) + \tan\theta \right]$$

For $\theta > \pi/2$

$$g(1) = \pm a_1 e^{ip_1} JJ_{11-} (0.5) \left[\tan(\alpha+\theta) (F_1-1) + \tan\theta \right] Q_1$$

$$g(3) = \pm a_3 e^{ip_3} JJ_{12-} (0.5) \left[\tan(\alpha+\theta) (F_3-1) + \tan\theta \right] Q_1.$$

The formulations of the basic scattering from Target ST-2 involved only first-order diffraction and were obtained by using the Ruck-Ufimtsev technique. Such higher order scattering as multiple diffraction among the edges, multiple reflection between the surfaces, or scattering mechanisms involving both a creep path along the surface and edge diffraction have not been modeled. However, by comparing the signatures computed from the basic formulation with wideband measurements data, the significance of the higher-order scattering mechanisms could be determined and a decision can be made concerning which is the better formulation of the basic scattering from the concaves edges.

L.6.1 Inputs

The subroutine input parameters are read from cards or passed in a common block. The parameters passed in the common block include:

NMIN = minimum frequency sample number

NMAX = maximum frequency sample number

DF = frequency increment (in MHz)

FC = carrier frequency (in GHz)

The card inputs are the following:

	MATH SYMBOL	VARIABLE	DEFINITION	COLUMNS
Card 1	θ	Aspect	Azimuth Angle (Degrees)	1-7
	a_1	A1	Radius of smallest cylinder (Inches)	8-14
	a_2	A2	Radius of middle cylinder (Inches)	15-21
	a_4	A4	Radius of largest cylinder (Inches)	22-28
	h_1	H1L	Length of smallest frustrum (Inches)	29-35
	h_2	H2L	Half-length of middle cylinder (Inches)	36-42
	h_3	H3L	Length of largest frustrum (Inches)	43-49
	h_4	H4L	Length of largest cylinder (Inches)	50-56
	h_5	H5L	Length of smallest cylinder (Inches)	57-63

L.6.2 Outputs

The data base output consists of four linear arrays, EVVR, EVVI, EHHR, EHHI, which contain the real and imaginary parts of the vertically and horizontally polarized back-scattered fields (in meters) at frequency increments of DF MHz from NMIN*DF to NMAX*DF.

L.6.3 Restrictions

L.6.3.1 Physical Dimensions

All target dimensions should be large with respect to the wavelength of the illuminating field. The target is further restricted such that two frustra form the same angle with the cylindrical surfaces, i. e., $\alpha = \tan^{-1} \frac{a_2 - a_1}{h_1} = \tan^{-1} \frac{a_4 - a_2}{h_3}$.

In addition, the basic shape should not be distorted by choosing α too close to 90 or 0 degrees. A value of $30 < \alpha < 80$ should be maintained.

In determining the shadowing of the various surfaces and edges the target geometry was further restricted such that $\alpha_3 < \alpha_4$ where these angles are defined by the equation. This restriction applies only to the use of the Q functions in determining the amplitude weighting of the scattered field terms due to shadowing.

L.6.3.2 Output

The output arrays are passed in the argument list and a value is computed and stored only in locations NMIN to NMAX.

L.6.3.3 Restrictions

The azimuth angle is restricted to the region between 0 and 180 degrees. In addition, the specular azimuths 0, 90, 180, $(90 - \alpha)$ should not be used. In order to compute the response at these angles, an angular offset of approximately 0.01 degrees should be used.

L.6.4 Definition of Selected Terms Used in Subroutine

$$D1 = \cos \pi/n$$

$$D11 = \frac{\sin \pi/n}{n}$$

$$E1 = \frac{1}{\cos \pi/n - 1}$$

where $n = n_1 = n_3 = 1 - \frac{\alpha}{\pi}$

$$\left\{\begin{matrix} V \\ H \end{matrix}\right\} 1 = C(n_1) \mp B(n_1, \theta) \qquad \text{for } \theta < \pi/2$$

$$PS1 = JJ_{11-} = JJ_{1m\mp} = \left[J_o(2ka_m \sin\theta) \mp i\, J_1(2ka_m \sin\theta) \right]$$

where m = 1 and the upper (-) sign is utilized

$$CC1 = C(n_1)\ JJ_{21}$$

$$CT1 = 0.5 \tan(\alpha + \theta)\ F_1$$

$$C\begin{Bmatrix} V \\ H \end{Bmatrix}1 = a_1 e^{ip_1}\Big\}\ \ JJ_{11} - \Big[C(n_1) \mp B(n_1,\theta) \pm 0.5 \tan(\alpha+\theta)F_1\Big]$$

$$- C(n_1)JJ_{21}\Big\{ \qquad \text{for } \theta < \pi/2$$

$$JJ_{21} = \Big[J_o(2ka_m \sin\theta) + J_2(2ka_m \sin\theta)\Big]$$

where m = 1

L.6.5 Subroutines Used

Subfunctions:

Q (X) returns the value of the exponential smoothing function

Subroutines:

1. BESL (XCx, Bx0, Bx1, Bx2) computes Bessel functions of first three orders for real argument Xcx and returns

   J_0 (XCx) in Bx0

   J_1 (XCx) in Bx1

   J_2 (XCx) in Bx2

2. DIFFC (VX,HX,NX,DX,DX1,EX,PHI) computes

   $$\begin{Bmatrix} VX \\ HX \end{Bmatrix} = C(N) \mp B(NX, PHI)$$

   The inputs are NX, DX, DX1, EX, and PHI and outputs are VX, and HX.

3. FTG (TSx, FTx) computes the Special F function using TSx as argument and returns value in FTx.

```
      SUBROUTINE TARGET  (EVVR,EVVI,EHHR,EHHI )
C     **   RESPONSE OF TARGET ST-2                                    **
C     **   COMPUTED UTILIZING THE RUCK-UFIMTSEV EQUATIONS   **
C
      COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, T0
C     NMIN = MINIMUM FREQUENCY SAMPLE
C     NMAX = MAXIMUM FREQUENCY SAMPLE
C     DF   = FREQUENCY INCREMENT IN MHZ
C     FC   = CARRIER FREQUENCY IN GHZ
      COMPLEX    PS1, PS2, PS4,PS1P,PS2P,PS4P,  C1,  C2,
     A  C4,  C3,  C9, C11, FT1,       FT2, FT4, FT5, FT6, FT7, FT8,
     B CT1, CT2, CT4, CT6, CT8, C1T, C3T, C2FT, C4FT
      COMPLEX    CV1, CH1, CV2, CH2, CV3, CH3, CV4, CH4, CV5, CH5,
     A CV6, CH6, CV7, CH7, CV8, CH8, CV9, CH9,CV10,CH10,CV11,CH11,
     BCV12,CH12,  CV,  CH, CVA, CHA
      REAL N1,N2,NC,ITM
      DIMENSION  EVVR(1), EVVI(1), EHHR(1), EHHI(1)
C
C
C
C     PROGRAM CONSTANTS
C
      PI = 3.14159265358979
      PI2 = PI+PI
      PIO2 = PI/2
      HPI  = .5*PI
      SPI  = SQRT(PI)
      RTD = 180./PI
      DTR = PI/180.
      SHPI = SQRT(HPI)
      SIHPI = 1./SHPI
      ITM =0.0254
      SMIN = 1.E-4
      SMDB = -80.
      C = 11.80285078
C
      READ(5,5001) ASPECT,A1,A2,A4,H1L,H2L,H3L,H4L,H5L,M1
 5001 FORMAT(9F7.2,I2)
      WRITE(6,5010) ASPECT
 5010 FORMAT (  29H1  PROGRAM INPUT PARAMETERS  ,/,
    1            17H  ASPECT ANGLE = , F9.4 )
      WRITE (6,6001) H1L,A1,H2L,A2,H3L,A4,H4L,H5L
 6001 FORMAT( //, 7H  H1 = ,F8.4 ,7H  A1 = ,F8.4,/  ,7H  H2 = ,F8.4 ,
     A   7H  A2 = ,F8.4,/  ,7H  H3 = ,F8.4 , 7H  A4 = ,F8.4,/  ,
     B   7H  H4 = ,F8.4,/  ,7H  H5 = ,F8.4  )
C
      TH = ASPECT*DTR
C
      H12L = H1L + H2L
      H125L = H1L + H2L + H5L
      H23L = H2L + H3L
      H234L = H2L + H3L + H4L
      A21 = A2-A1
      A42 = A4-A2
C
      X1 = ATAN( A21/H1L)
      X2 = X1
      X3 = ATAN( A42/(H2L+H23L))
      X4 = ATAN( A21/(H1L+H5L))

      PIMX3 = PI-X3
      PIMX4 = PI-X4

      X1D = X1*RTD
      X2D = X2*RTD
      X3D = X3*RTD
      X4D = X4*RTD

      WRITE (6,6005) X1D,X2D,X3D,X4D
 6005 FORMAT ( ///, 8H  X1D = ,F7.3, 8H  X2D = ,F7.3, / ,
     A                8H  X3D = ,F7.3, 8H  X4D = ,F7.3  )

      N1 = 1.-X1/PI
      N2 = 1.+X1/PI
      NC = 1.5

      D1 = COS(PI/N1)
      D2 = COS(PI/N2)
      DC = -.5

      D11 = 1./(D1-1.)
      D21 = 1./(D2-1.)
      DC1 = -2./3.

      E1 = SIN(PI/N1)/N1
      E2 = SIN(PI/N2)/N2
      EC = SIN(PI/NC)/NC

      STH = SIN(TH)
      CTH = COS(TH)

      X0 =(2.*PI)/C
      AX0=(X0/1000.)
      XAK2 = X0*FC
      AK2A1 =XAK2*A1
      AK2A2 =XAK2*A2
      AK2A4 =XAK2*A4

      XPT  = X1+TH
      C0 = TAN(XPT)*0.5
      TSP = COS(XPT)/SIN(X1)
      IF ( TH .GT. PIO2) GO TO 20
      XMT = X1-TH
      CK0 = 0.5+TAN(XMT)
      TSM = COS(XMT)/SIN(X1)
      Z6 = AK2A2*(X4-TH)
      Z8 = AK2A4*(X3-TH)
      Q8= Q(Z8)
      Q6= Q(Z6)
      DIFFC COMPUTES  C(N)-/+B(N,PHI) TERMS RETURNED AS VX,HX
      CALL DIFFC( V12, H12,NC,DC,DC1,EC, PIO2-TH )
      CALL DIFFC ( V11,H11 ,NC,DC,DC1,EC, PIO2+TH )
      CALL DIFFC ( V1,H1, N1,D1,D11,E1,TH )
      CALL DIFFC ( V2,H2, N2,D2,D21,E2, XPT  )
      CALL DIFFC ( V9,H9 ,NC,DC,DC1,EC,  TH )
      CALL DIFFC ( V6,H6,  N2,D2,D21,E2,XMT )
      GO TO 30
   20 CONTINUE
      PIMTH = PI-TH
      Z2 = AK2A2*(PIMX3-TH)
      PIMXPT = PI-X1-TH
      Z1 =   AK2A1*PIMXPT*3.
      Z11= AK2A1*(PIMX4-TH)
      Q11= Q(Z11)
      Q2 = Q(Z2)
      Q1 = Q(Z1)
C
      CALL DIFFC ( V9,H9,NC,DC,DC1,EC,  TH )
      CALL DIFFC ( V10,H10,NC,DC,DC1,EC,  TH-PIO2 )
      CALL DIFFC ( V4,H4 ,N2,D2,D21,E2,  PIMTH )
      CALL DIFFC ( V1,H1,  N1,D1,D11,E1,PIMXPT)
      CALL DIFFC(V11,H11,NC,DC,DC1,EC,  PIMTH )
   30 CONTINUE
      DO 50  I= NMIN,NMAX
      XI = I-1
      AK =AX0*XI*DF
      AK2 = 2.0*AK
      AK2A1 = AK2*A1
      AK2A2 = AK2*A2
      AK2A4 = AK2*A4
C
      XC1=AK2A1*STH
      XC2=AK2A2*STH
      XC4=AK2A4*STH
C
      CALL BESL(XC1,B10,B11,B12)
      CALL BESL(XC2,B20,B21,B22)
      CALL BESL(XC4,B40,B41,B42)
C
C     PHASE AND AMPLITUDE TERMS FROM BESSEL FUNCTIONS
      PS1 = CMPLX( B10,-B11)
      PS2 = CMPLX( B20,-B21)
      PS4 = CMPLX( B40,-B41)
      PS1P = CONJG(PS1)
      PS2P = CONJG(PS2)
      PS4P = CONJG(PS4)
C
C     PHASE TERM USING LENGTH ALONG AXIS
      PC1 = -AK2*H12L*CTH
      PC2 = -AK2*H2L*CTH
      PC4 = AK2*H23L*CTH
      PC9 = AK2*H234L*CTH
      PC11=-AK2*H125L*CTH
C
      C1    = CMPLX(COS(PC1),SIN(PC1))
      C2    = CMPLX(COS(PC2),SIN(PC2))
      C4    = CMPLX(COS(PC4),SIN(PC4))
      C3    = CONJG(  C2)
      C9    = CMPLX(COS(PC9),SIN(PC9))
      C11   =  CMPLX(COS(PC11),SIN(PC11))
C
      C1 = A1*SPI*C1
```

```
C3 = A2*SPI*C3
C2 = A2*SPI*C2
C4 = A4*SPI*C4
C9 = A4*SPI*C9
C11 = A1*SPI*C11

CAUSTIC CORRECTION TERMS
CC11 = E0*D01*(B10+B12)
CC1  = E1*D11*(B10+B12)
CC2  = E2*D21*(B20+B22)
CC3  = E1*D11*(B20+B22)
CC4  = E2*D21*(B40+B42)
CC9  = E0*D01*(B40+B42)

TSPP = TSP*AK2
TS1= A1*TSPP
TS2= A2*TSPP
TS4= A4*TSPP

IF ( TH .GT. PIO2 ) GO TO 450


CV12 = C11*(V12*PS1P - CC11)
CH12 = C11*(H12*PS1P - CC11)
CV11 = C11*(V11*PS1 - CC11)
CH11 = C11*(H11*PS1 - CC11)

CALL FTG (TS1,FT1)
CT1 = C0*FT1
CV1 = C1*((V1+CT1)*PS1 - CC1)
CH1 = C1*((H1-CT1)*PS1 - CC1)

CALL FTG (TS2,FT2)
CT2 = C0*FT2
CV3 = C3*((V1+CT2)*PS2 - CC3)
CH3 = C3*((H1-CT2)*PS2 - CC3)

CV2 = C2*((V2-CT2)*PS2 - CC2)
CH2 = C2*((H2+CT2)*PS2 - CC2)

CALL FTG (TS4,FT4)
CT4 = C0*FT4
CV4 = C4*((V2-CT4)*PS4 - CC4)
CH4 = C4*((H2+CT4)*PS4 - CC4)

CV9 = C9*(V9*PS4 - CC9)
CH9 = C9*(H9*PS4 - CC9)

CV = CV11+CV12+CV1+CV2+CV3+CV4+CV9
CH = CH11+CH12+CH1+CH2+CH3+CH4+CH9

IF ( Z6 .LE.-2. )GO TO 801

TSMM = TSM*AK2
TS5 =TSMM*A1
TS6 =TSMM*A2
TS8 =TSMM*A4
```

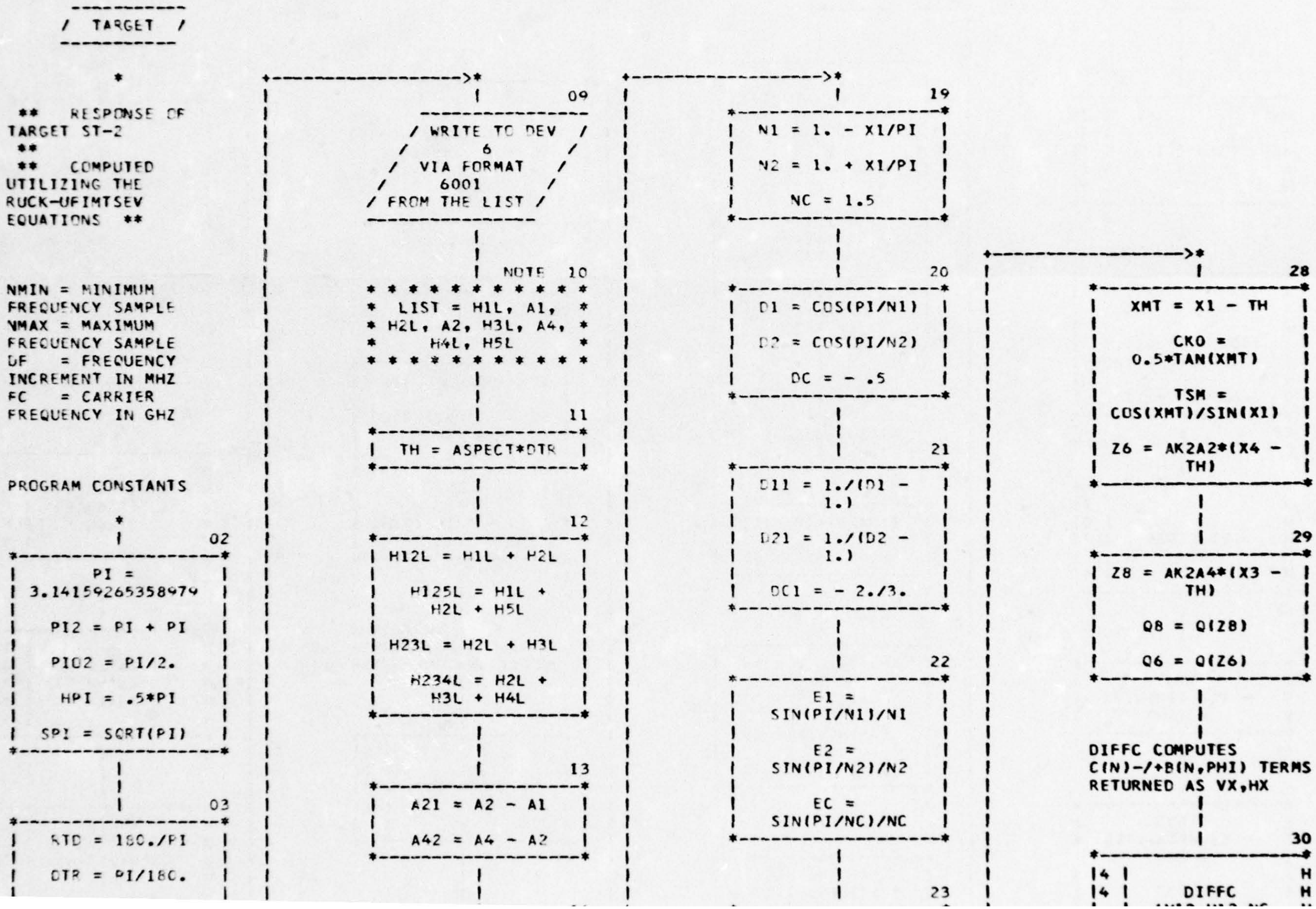
```
      CALL FTG(TS6,FT6)
      CT6 = CK0*FT6
      CV6 = C2*((V6-CT6)*PS2P - CC2)*Q6
      CH6 = C2*((H6+CT6)*PS2P - CC2)*Q6

      CALL FTG(TS5,FT5)
      FT5 = CK0*(1.-FT5)*PS1P*C1*Q6
      CV5 = -FT5
      CH5 = +FT5

      FT7 = CK0*(1.-FT6)*PS2P*C3*Q8
      CV7 = -FT7
      CH7 = FT7

      CALL FTG(TS8,FT8)
      CT8 = CK0*FT8
      CV8 = C4*((V6-CT8)*PS4F - CC4)*Q8
      CH8 = C4*((H6+CT8)*PS4P - CC4)*Q8

      CVA = CV5 + CV6 + CV7 + CV8
      CHA = CH5 + CH6 + CH7 + CH8

      CV = CV+CVA
      CH = CH+CHA
      GO TO 801

  456 CONTINUE

      THETA GREATER THAN PI/2

      CV9 = C9*(V9*PS4 - CC9)
      CH9 = C9*(H9*PS4 - CC9)

      CV10 = C9*(V10*PS4P - CC9)
      CH10 = C9*(H10*PS4P - CC9)

      CV4 = C4*    ( V4*PS4 - CC4)
      CH4 = C4*    ( H4*PS4 - CC4)
      CV = CV9 + CV10 +CV4
      CH = CH9 + CH10 +CH4

      IF (    Z2        .LE. -2. ) GO TO 800

      CV11 = C11* (V11*PS1 - CC11)*Q11
      CH11 = C11* (H11*PS1 - CC11)*Q11

      CV2 =    C2* (V4*PS2 - CC2)*Q2
      CH2 =    C2* (H4*PS2 - CC2)*Q2

      CV = CV + CV11 + CV2
      CH = CH + CH11 + CH2
      IF (    Z1 .GT. -2. ) GO TO 700
      HTTH = 0.5*(STH/CTH)
      C1T = C1*HTTH*PS1*Q11
      C3T = C3*HTTH*PS2*Q2
      CV = CV + C1T + C3T
      CH = CH - C1T - C3T
      GO TO 800
C
  700 CONTINUE
C
      CALL FTG (TS1,FT1 )
      CV1 = C1*    ((V1+C0*FT1)*PS1 - CC1 )
      CH1 = C1*    ((H1-C0*FT1)*PS1 - CC1 )
      CALL FTG (TS2,FT2 )
      CV3 = C3*    ((V1+C0*FT2)*PS2 - CC3 )
      CH3 = C3*    ((H1-C0*FT2)*PS2 - CC3 )
C
      C2FT = C2    *C0*FT2*PS2
      CALL FTG (TS4,FT4)
      C4FT = C4    *C0*FT4*PS4
C
      CV = CV +(CV1 + CV3 - C2FT - C4FT)*01
      CH = CH +(CH1 + CH3 + C2FT + C4FT)*01
C
C
  800 CONTINUE
C
  801 CONTINUE
      CV =-CV*ITM
      CH = CH*ITM
      EVVR(I) = REAL(CV)
      EVVI(I) = -AIMAG(CV)
      EHHR(I) = REAL(CH)
      EHHI(I) = -AIMAG(CH)
   50 CONTINUE
      RETURN
      END
      SUBROUTINE DIFFC ( V,H,N,D,D1,E, PHI )
      REAL N
C
      D2 = 1./(D-COS((PHI+PHI)/N))
      V  = E*(D1-D2)
      H  = E*(D1+D2)
      RETURN
      END
      SUBROUTINE BESL ( X, B0, B1, B2 )
C
C   * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS
C   * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS
C   * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )
C
C
      S = 1.0
      IF ( X.LT. 0. ) S= -1.
      X = ABS(X)
      IF ( X .GT. 1.E-6 ) GO TO 5
      B0 = 1.0
      B1 = 0.0
      B2 = 0.0
      X = X*S
      RETURN
C
    5 CONTINUE
```

```
C
    1 IF ( X .GE. 3. ) GO TO 9
      X1 = X/3.
      X1 = X1*X1
      B = 1. + X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479
     1 +        X1*(-.0039444+ X1*2.1E-4 )))) )
      GO TO 10
C
    9 X2 = 3./X
      F0 =  .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*
     1      (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )
      T0 = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573
     1     +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))
      B = F0*COS(T0)/SQRT(X)
C
   10 B0 = B
C
    2 IF ( X .GE. 3. ) GO TO 19
      X1 = X/3.
      X1 = X1*X1
      B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*
     1                 (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )
      GO TO 20
C
   19 X2 = 3./X
      F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*
     1     (-.00249511 +X2*(.00113653 -.00020033*X2 )))))
      T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879
     1     +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))
      B = F1*COS(T1)/SQRT(X)
C
   20 B1 = B*S
      X = X*S
      B2= (2./X)*B1 - B0
   50 RETURN
      END
      SUBROUTINE FTG(TAUS,F)
C
C     COMPUTES FTAU WHERE FTAU =(EXP(-J*TAU**2)/2*TAU)*SQRT(PI/2.)*
C                               (C2(TAU**2) + J*S2(TAU**2))
C
      COMPLEX F,FP
      PI = 3.14159265358979
      PIO2 = PI/2.
      C1 = SQRT(PI/2.)
      C2 = 1./C1
      ATAUS = ABS(TAUS)
      IF (ATAUS .LE. 0.5 )GO TO 20
C
C     FOR TAUS .GT. 0.5, FUNCTION COMPUTED USING POLYNOMIAL
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,
C   *               SECTIONS  7.3.9,7.3.10,7.3.32,7.3.33)
      TAU = SQRT(ATAUS)
      X = C2*TAU
      XS = X*X
C
      FX = (1.0+0.926*X)/(2.0+1.792*X+3.104*XS)
      GX = 1.0/(2.0+4.142*X+3.492*XS+6.67*X*XS)
```

```
C
      CC1XS = COS(ATAUS)
      SC1XS = SIN(ATAUS)
C
      CX = 0.5 + FX*SC1XS - GX*CC1XS
      SX = 0.5 - FX*CC1XS - GX*SC1XS
C
      IF (TAUS .LT. 0.) GO TO 10
      F= CMPLX( CX, SX)
      FP = CMPLX(  COS(ATAUS), -SIN(ATAUS) )
      F =(C1*F*FP)/TAU
      RETURN
C
   10 CONTINUE
      F = CMPLX(SX, CX)
      A = ATAUS-PIO2
      FP = CMPLX( COS(A), SIN(A) )
      F = (F*FP*C1)/TAU
      RETURN
C
C     FOR TAUS .LE. 0.5, FUNCTION IS EXPANDED IN SERIES AND FIRST FEW
C                        TERMS INTEGRATED TERM BY TERM TO OBTAIN RESULT
   20 CONTINUE
      FP = CMPLX( COS(TAUS),-SIN(TAUS) )
      TS = TAUS*TAUS
      FR = 1 - TS*( .1 - .0046296296*TS)
      FI = TAUS*( .333333333 - TS*(.0238095238 - 7.57575757E-4*TS))
      F = CMPLX( FR,FI )
      F = FP*F
      RETURN
      END
      FUNCTION Q(Z)
C     Q(Z) = 0.5*(1 + ERF(Z))
C   * ERF(Z) IS EVALUATED USING A RATIONAL POLYNOMIAL APPROXIMATION
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,
C   *              SECTION 7.1.26)
C
      IF ( Z.GT. 2.) GO TO 10
      IF ( Z.LT.-2.) GO TO 20
      AZ = ABS(Z)
      P = 1.0/(1.0 + .47047*AZ)
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)
      IF (Z) 2,4,6
    2 Q = (1.0 - Y)/2.
      RETURN
    4 Q = .5
      RETURN
    6 Q = (1.0 + Y)/2.
      RETURN
   10 Q = 1.
      RETURN
   20 Q = 0.
      RETURN
      END
```

04/26/76 AUTOFLOW CHART SET - FWO/SCL RADSIM

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI)



```
/ TARGET /

**   RESPONSE OF
TARGET ST-2
**
**   COMPUTED
UTILIZING THE
RUCK-UFIMTSEV
EQUATIONS  **

NMIN = MINIMUM
FREQUENCY SAMPLE
NMAX = MAXIMUM
FREQUENCY SAMPLE
DF   = FREQUENCY
INCREMENT IN MHZ
FC   = CARRIER
FREQUENCY IN GHZ

PROGRAM CONSTANTS

02
PI =
3.14159265358979
PI2 = PI + PI
PIO2 = PI/2.
HPI = .5*PI
SPI = SQRT(PI)

03
RTD = 180./PI
DTR = PI/180.
```

```
09
/ WRITE TO DEV /
/ 6 /
/ VIA FORMAT /
/ 6001 /
/ FROM THE LIST /

NOTE 10
* * * * * * * * * * *
* LIST = H1L, A1,   *
* H2L, A2, H3L, A4, *
*   H4L, H5L        *
* * * * * * * * * * *

11
TH = ASPECT*DTR

12
H12L = H1L + H2L
H125L = H1L +
  H2L + H5L
H23L = H2L + H3L
H234L = H2L +
  H3L + H4L

13
A21 = A2 - A1
A42 = A4 - A2
```

```
19
N1 = 1. - X1/PI
N2 = 1. + X1/PI
NC = 1.5

20
D1 = COS(PI/N1)
D2 = COS(PI/N2)
DC = - .5

21
D11 = 1./(D1 -
  1.)
D21 = 1./(D2 -
  1.)
DC1 = - 2./3.

22
E1 =
SIN(PI/N1)/N1
E2 =
SIN(PI/N2)/N2
EC =
SIN(PI/NC)/NC

23
```

```
28
XMT = X1 - TH
CKO =
0.5*TAN(XMT)
TSM =
COS(XMT)/SIN(X1)
Z6 = AK2A2*(X4 -
  TH)

29
Z8 = AK2A4*(X3 -
  TH)
Q8 = Q(Z8)
Q6 = Q(Z6)

DIFFC COMPUTES
C(N)-/+B(N,PHI) TERMS
RETURNED AS VX,HX

30
4 | DIFFC   H
4 |         H
```

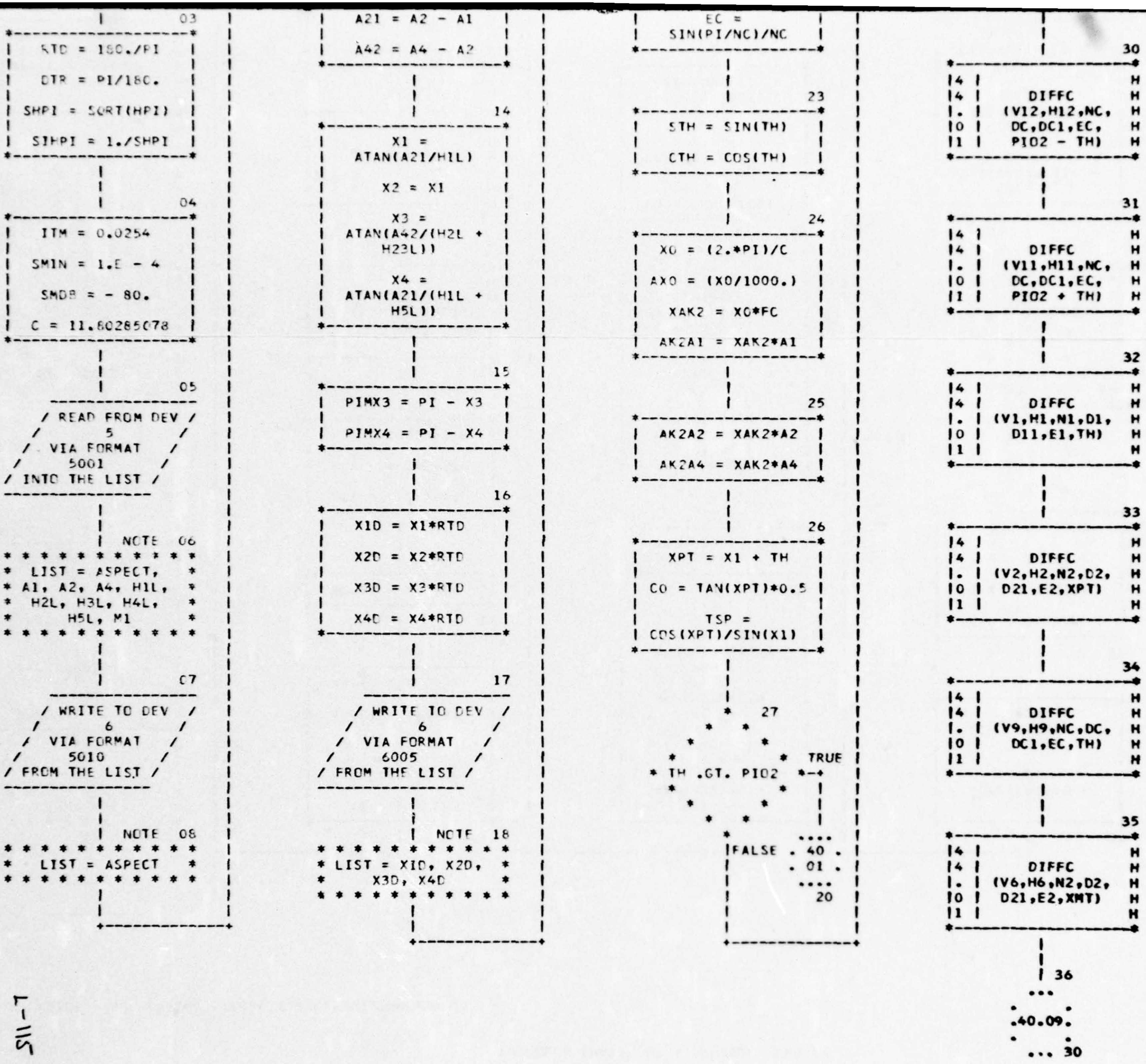
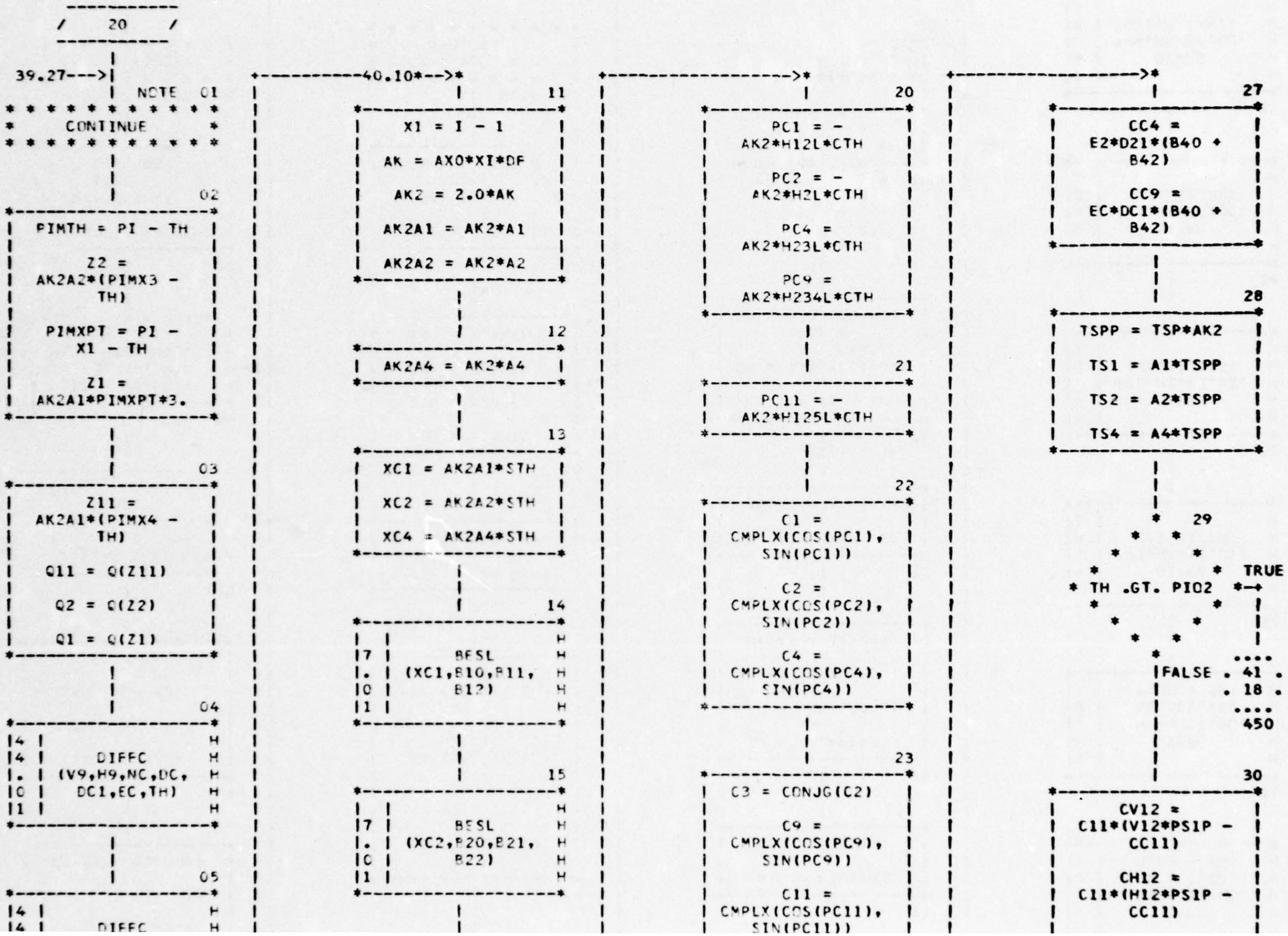

03
STD = 180./PI
DTR = PI/180.
SHPI = SQRT(HPI)
SIHPI = 1./SHPI
04
ITM = 0.0254
SMIN = 1.E - 4
SMDB = - 80.
C = 11.80285078
05
READ FROM DEV 5 VIA FORMAT 5001 INTO THE LIST
NOTE 06
LIST = ASPECT, A1, A2, A4, H1L, H2L, H3L, H4L, H5L, M1
07
WRITE TO DEV 6 VIA FORMAT 5010 FROM THE LIST
NOTE 08
LIST = ASPECT
A21 = A2 - A1
A42 = A4 - A2
14
X1 = ATAN(A21/H1L)
X2 = X1
X3 = ATAN(A42/(H2L + H23L))
X4 = ATAN(A21/(H1L + H5L))
15
PIMX3 = PI - X3
PIMX4 = PI - X4
16
X1D = X1*RTD
X2D = X2*RTD
X3D = X3*RTD
X4D = X4*RTD
17
WRITE TO DEV 6 VIA FORMAT 6005 FROM THE LIST
NOTE 18
LIST = X1D, X2D, X3D, X4D
EC = SIN(PI/NC)/NC
23
STH = SIN(TH)
CTH = COS(TH)
24
X0 = (2.*PI)/C
AX0 = (X0/1000.)
XAK2 = X0*FC
AK2A1 = XAK2*A1
25
AK2A2 = XAK2*A2
AK2A4 = XAK2*A4
26
XPT = X1 + TH
C0 = TAN(XPT)*0.5
TSP = COS(XPT)/SIN(X1)
27
TH .GT. PIO2
TRUE
FALSE
40
01
20
30
44.01
DIFFC (V12,H12,NC, DC,DC1,EC, PIO2 - TH)
HHHHH
31
44.01
DIFFC (V11,H11,NC, DC,DC1,EC, PIO2 + TH)
HHHHH
32
44.01
DIFFC (V1,H1,N1,D1, D11,E1,TH)
HHHHH
33
44.01
DIFFC (V2,H2,N2,D2, D21,E2,XPT)
HHHHH
34
44.01
DIFFC (V9,H9,NC,DC, DC1,EC,TH)
HHHHH
35
44.01
DIFFC (V6,H6,N2,D2, D21,E2,XMT)
HHHHH
36
40.09
30

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,FHHR,FHHI)

L-116



```
/ 20 /
39.27--->
NOTE 01
CONTINUE
02
PIMTH = PI - TH
Z2 = AK2A2*(PIMX3 - TH)
PIMXPT = PI - X1 - TH
Z1 = AK2A1*PIMXPT*3.
03
Z11 = AK2A1*(PIMX4 - TH)
Q11 = Q(Z11)
Q2 = Q(Z2)
Q1 = Q(Z1)
04
4 4 . 0 1   DIFFC (V9,H9,NC,DC, DC1,EC,TH)
05
4 4   DIFFC

40.10*-->*
11
X1 = I - 1
AK = AX0*XI*DF
AK2 = 2.0*AK
AK2A1 = AK2*A1
AK2A2 = AK2*A2
12
AK2A4 = AK2*A4
13
XC1 = AK2A1*STH
XC2 = AK2A2*STH
XC4 = AK2A4*STH
14
7 . 0 1   BESL (XC1,B10,B11, B12)
15
7 . 0 1   BESL (XC2,B20,B21, B22)

20
PC1 = - AK2*H12L*CTH
PC2 = - AK2*H2L*CTH
PC4 = AK2*H23L*CTH
PC9 = AK2*H234L*CTH
21
PC11 = - AK2*H125L*CTH
22
C1 = CMPLX(COS(PC1), SIN(PC1))
C2 = CMPLX(COS(PC2), SIN(PC2))
C4 = CMPLX(COS(PC4), SIN(PC4))
23
C3 = CONJG(C2)
C9 = CMPLX(COS(PC9), SIN(PC9))
C11 = CMPLX(COS(PC11), SIN(PC11))

27
CC4 = E2*D21*(B40 + B42)
CC9 = EC*DC1*(B40 + B42)
28
TSPP = TSP*AK2
TS1 = A1*TSPP
TS2 = A2*TSPP
TS4 = A4*TSPP
29
TH .GT. PIO2
TRUE
FALSE
41 . 18
450
30
CV12 = C11*(V12*PSIP - CC11)
CH12 = C11*(H12*PSIP - CC11)
```

2

```
|4 |      DIFFC       H  |
|. | (V9,H9,NC,DC,    H  |
|0 |   DC1,EC,TH)     H  |
|1 |                  H  |
*------------------------*  |
            |               |
            |               |
            |         05    |
*------------------------*  |
|4 |                  H  |
|4 |      DIFFC       H  |
|. | (V10,H10,NC,     H  |
|0 | DC,DC1,EC,TH -   H  |
|1 |      PIO2)       H  |
*------------------------*  |
            |               |
            |               |
            |         06    |
*------------------------*  |
|4 |                  H  |
|4 |      DIFFC       H  |
|. | (V4,H4,N2,D2,    H  |
|0 | D21,E2,PIMTH)    H  |
|1 |                  H  |
*------------------------*  |
            |               |
            |               |
            |         C7    |
*------------------------*  |
|4 |                  H  |
|4 |      DIFFC       H  |
|. | (V1,H1,N1,D1,    H  |
|0 | D11,E1,PIMXPT)   H  |
|1 |                  H  |
*------------------------*  |
            |               |
            |               |
            |         C8    |
*------------------------*  |
|4 |                  H  |
|4 |      DIFFC       H  |
|. | (V11,H11,NC,     H  |
|0 |  DC,DC1,EC,      H  |
|1 |    PIMTH)        H  |
*------------------------*  |
            |               |
 39.36--->  |               |
    30      |  NOTE  09     |
* * * * * * * * * * * * *   |
*        CONTINUE       *   |
* * * * * * * * * * * * *   |
            |               |
            |  NOTE  10     |
* * * * * * * * * * * * *   |
*     BEGIN DO LOOP     *   |
* 50 I = NMIN, NMAX     *   |
* * * * * * * * * * * * *   |
            |               |
            |               |
            +---------------+
```

```
             |                 |
             |          15     |
*------------------------*     |
|  |                    H      |
|7 |       BESL         H      |
|. | (XC2,B20,B21,      H      |
|0 |      B22)          H      |
|1 |                    H      |
*------------------------*     |
             |                 |
             |                 |
             |          16     |
*------------------------*     |
|  |                    H      |
|7 |       BESL         H      |
|. | (XC4,B40,B41,      H      |
|0 |      B42)          H      |
|1 |                    H      |
*------------------------*     |
             |                 |
             |                 |
PHASE AND AMPLITUDE            |
TERMS FROM BESSEL              |
FUNCTIONS                      |
             |                 |
             |                 |
             |          17     |
*------------------------*     |
|        PS1 =           |     |
| CMPLX(B10, - B11)      |     |
|                        |     |
|        PS2 =           |     |
| CMPLX(B20, - B21)      |     |
|                        |     |
|        PS4 =           |     |
| CMPLX(B40, - B41)      |     |
|                        |     |
| PS1P = CONJG(PS1)      |     |
*------------------------*     |
             |                 |
             |                 |
             |          18     |
*------------------------*     |
| PS2P = CONJG(PS2)      |     |
|                        |     |
| PS4P = CONJG(PS4)      |     |
*------------------------*     |
             |                 |
             |                 |
PHASE TERM USING               |
LENGTH ALONG AXIS              |
             |                 |
             |                 |
             +-----------------+
```

```
              |                 |
              |          23     |
*------------------------*      |
| C3 = CONJG(C2)         |      |
|                        |      |
|        C9 =            |      |
| CMPLX(COS(PC9),        |      |
|     SIN(PC9))          |      |
|                        |      |
|        C11 =           |      |
| CMPLX(COS(PC11),       |      |
|     SIN(PC11))         |      |
*------------------------*      |
              |                 |
              |          24     |
*------------------------*      |
| C1 = A1*SPI*C1         |      |
|                        |      |
| C3 = A2*SPI*C3         |      |
|                        |      |
| C2 = A2*SPI*C2         |      |
|                        |      |
| C4 = A4*SPI*C4         |      |
*------------------------*      |
              |                 |
              |                 |
              |          25     |
*------------------------*      |
| C9 = A4*SPI*C9         |      |
|                        |      |
| C11 = A1*SPI*C11       |      |
*------------------------*      |
              |                 |
              |                 |
CAUSTIC CORRECTION              |
TERMS                           |
              |                 |
              |                 |
              |          26     |
*------------------------*      |
|       CC11 =           |      |
|  EC*DC1*(B10 +         |      |
|       B12)             |      |
|                        |      |
|       CC1 =            |      |
|  E1*D11*(B10 +         |      |
|       B12)             |      |
|                        |      |
|       CC2 =            |      |
|  E2*D21*(B20 +         |      |
|       B22)             |      |
|                        |      |
|       CC3 =            |      |
|  E1*D11*(B20 +         |      |
|       B22)             |      |
*------------------------*      |
              |                 |
              |                 |
              +-----------------+
```

```
              |
              |          30
*------------------------*
|       CV12 =           |
| C11*(V12*PS1P -        |
|       CC11)            |
|                        |
|       CH12 =           |
| C11*(H12*PS1P -        |
|       CC11)            |
|                        |
|       CV11 =           |
| C11*(V11*PS1 -         |
|       CC11)            |
*------------------------*
              |
              |          31
*------------------------*
|       CH11 =           |
| C11*(H11*PS1 -         |
|       CC11)            |
*------------------------*
              |
              |
              |          32
*------------------------*
|4 |                    H
|7 |       FTG          H
|. |    (TS1,FT1)       H
|0 |                    H
|2 |                    H
*------------------------*
              |
              |
              |          33
*------------------------*
|   CT1 = CO*FT1         |
|                        |
|  CV1 = C1*((V1 +       |
|  CT1)*PS1 - CC1)       |
|                        |
|  CH1 = C1*((H1 -       |
|  CT1)*PS1 - CC1)       |
*------------------------*
              |
              |
              |          34
*------------------------*
|4 |                    H
|7 |       FTG          H
|. |    (TS2,FT2)       H
|0 |                    H
|2 |                    H
*------------------------*
              |
              |
              |  /
           -------
           /41.01
```

```
04/26/76                                                  AUTOFLOW CHART SET - FWD/SCL  RADSIM

   CHART TITLE - SUBROUTINE  TARGET(EVVR,EVVI,EHHR,EHHI)


                                                                   -----------
                                                                  /   450    /
                                                                 -----------
                                                                       |
 40.35--->*                                                   40.29--->|
          |         01                                                 | NOTE  18
*-------------------*                                         * * * * * * * * * * *
|  CT2 = CO*FT2     |                                         *      CONTINUE      *
|                   |      *----------------------->*         * * * * * * * * * * *
|  CV3 = C3*((V1 +  |      |                        |      09            |
|  CT2)*PS2 - CC3)  |      |         *-------------------*               |
|                   |      |         |4 |               H                |
|  CH3 = C3*((H1 -  |      |         |7 |     FTG       H         THETA GREATER THAN
|  CT2)*PS2 - CC3)  |      |         |. |  (TS6,FT6)    H         PI/2
*-------------------*      |         |0 |               H                |
          |                |         |2 |               H                |
          |                |         *-------------------*               |          19
          |         02     |                  |                  *-------------------*
*-------------------*      |                  |                  |      CV9 =        |
|  CV2 = C2*((V2 -  |      |                  |      10          | C9*(V9*PS4 - CC9) |
|  CT2)*PS2 - CC2)  |      |         *-------------------*       |                   |
|                   |      |         |  CT6 = CKO*FT6    |       |      CH9 =        |
|  CH2 = C2*((H2 +  |      |         |                   |       | C9*(H9*PS4 - CC9) |
|  CT2)*PS2 - CC2)  |      |         |  CV6 = C2*((V6 -  |       *-------------------*
*-------------------*      |         |   CT6)*PS2P -     |                 |
          |                |         |    CC2)*Q6        |                 |
          |                |         |                   |                 |          20
          |         03     |         |  CH6 = C2*((H6 +  |       *-------------------*
*-------------------*      |         |   CT6)*PS2P -     |       |      CV10 =       |
|4 |                H      |         |    CC2)*Q6        |       | C9*(V10*PS4P -    |
|7 |     FTG        H      |         *-------------------*       |      CC9)         |
|. |  (TS4,FT4)     H      |                  |                  |                   |
|0 |                H      |                  |                  |      CH10 =       |
|2 |                H      |                  |      11          | C9*(H10*PS4P -    |
*-------------------*      |         *-------------------*       |      CC9)         |
          |                |         |4 |               H        *-------------------*
          |                |         |7 |     FTG       H                  |
          |         04     |         |. |  (TS5,FT5)    H                  |
*-------------------*      |         |0 |               H                  |          21
|  CT4 = CO*FT4     |      |         |2 |               H        *-------------------*
|                   |      |         *-------------------*       |      CV4 =        |
|  CV4 = C4*((V2 -  |      |                  |                  | C4*(V4*PS4 - CC4) |
|  CT4)*PS4 - CC4)  |      |                  |                  |                   |
|                   |      |                  |      12          |      CH4 =        |
|  CH4 = C4*((H2 +  |      |         *-------------------*       | C4*(H4*PS4 - CC4) |
|  CT4)*PS4 - CC4)  |      |         | FT5 = CKO*(1. -   |       |                   |
*-------------------*      |         | FT5)*PS1P*C1*Q6   |       | CV = CV9 + CV10 + |
          |                |         |                   |       |      CV4          |
          |                |         |   CV5 = - FT5     |       |                   |
          |         05     |         |                   |       | CH = CH9 + CH10 + |
*-------------------*      |         |   CH5 = + FT5     |       |      CH4          |
```

,2

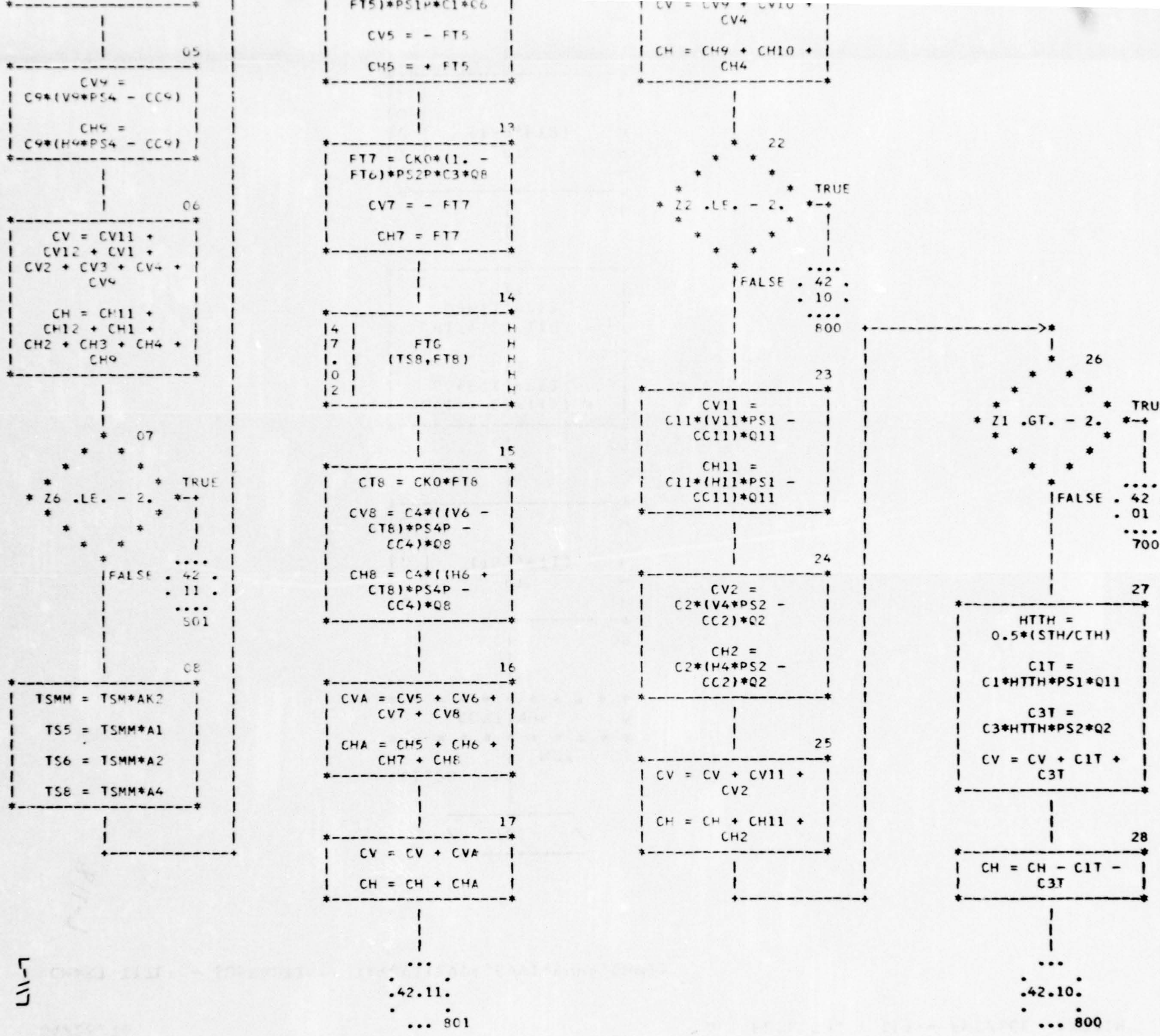
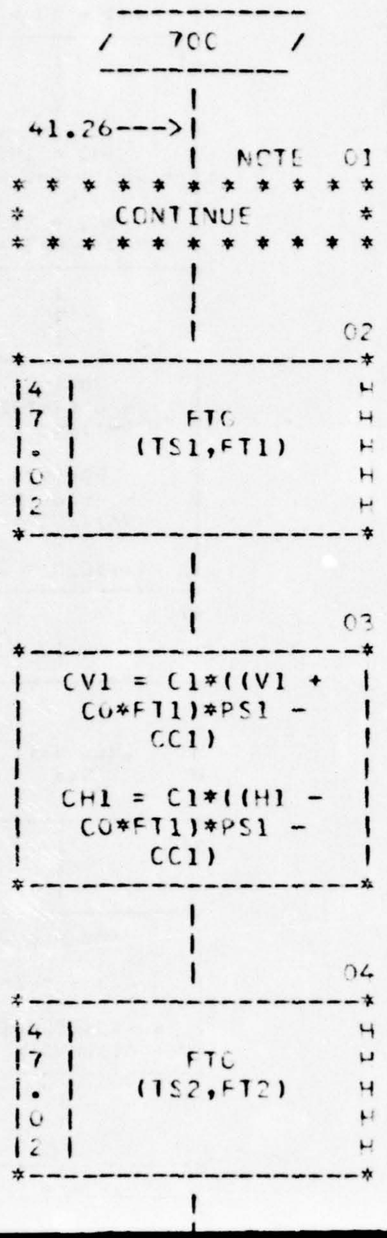

FT5)*PS1P*C1*C6
CV5 = - FT5
CH5 = + FT5
13
FT7 = CK0*(1. -
FT6)*PS2P*C3*Q8
CV7 = - FT7
CH7 = FT7
14
4
7
.
0
2
FTG
(TS8,FT8)
15
CT8 = CK0*FT8
CV8 = C4*((V6 -
CT8)*PS4P -
CC4)*Q8
CH8 = C4*((H6 +
CT8)*PS4P -
CC4)*Q8
16
CVA = CV5 + CV6 +
CV7 + CV8
CHA = CH5 + CH6 +
CH7 + CH8
17
CV = CV + CVA
CH = CH + CHA
.42.11.
... 801
05
CV9 =
C9*(V9*PS4 - CC9)
CH9 =
C9*(H9*PS4 - CC9)
06
CV = CV11 +
CV12 + CV1 +
CV2 + CV3 + CV4 +
CV9
CH = CH11 +
CH12 + CH1 +
CH2 + CH3 + CH4 +
CH9
07
Z6 .LE. - 2.
TRUE
FALSE
. 42 .
. 11 .
501
08
TSMM = TSM*AK2
TS5 = TSMM*A1
TS6 = TSMM*A2
TS8 = TSMM*A4
CV = CV9 + CV10 +
CV4
CH = CH9 + CH10 +
CH4
22
Z2 .LE. - 2.
TRUE
FALSE
. 42 .
. 10 .
800
23
CV11 =
C11*(V11*PS1 -
CC11)*Q11
CH11 =
C11*(H11*PS1 -
CC11)*Q11
24
CV2 =
C2*(V4*PS2 -
CC2)*Q2
CH2 =
C2*(H4*PS2 -
CC2)*Q2
25
CV = CV + CV11 +
CV2
CH = CH + CH11 +
CH2
26
Z1 .GT. - 2.
TRUE
FALSE
. 42 .
. 01 .
700
27
HTTH =
0.5*(STH/CTH)
C1T =
C1*HTTH*PS1*Q11
C3T =
C3*HTTH*PS2*Q2
CV = CV + C1T +
C3T
28
CH = CH - C1T -
C3T
.42.10.
... 800

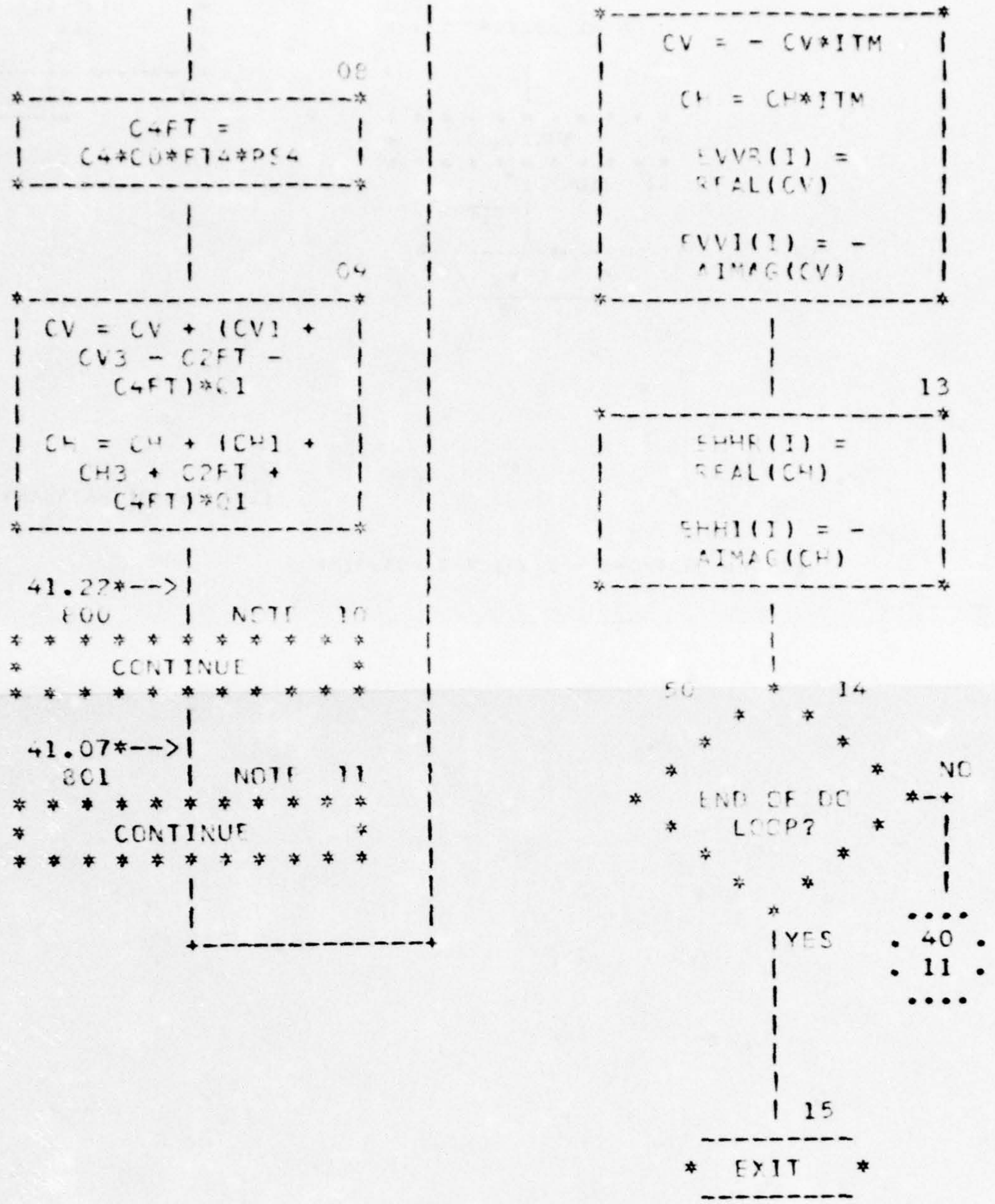
CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI)

L-118



```
|.  |   (TS2,FT2)      H
|0  |                  H
|2  |                  H
*----------------------*
           |
           |
           |          05
*----------------------*
|  CV3 = C3*((V1 +     |
|   C0*FT2)*PS2 -      |
|       CC3)           |
|                      |
|  CH3 = C3*((H1 -     |
|   C0*FT2)*PS2 -      |
|       CC3)           |
*----------------------*
           |
           |
           |          06
*----------------------*
|       C2FT =         |
|   C2*C0*FT2*PS2      |
*----------------------*
           |
           |
           |          07
*----------------------*
|4  |                  H
|7  |      FTC         H
|.  |   (TS4,FT4)      H
|0  |                  H
|2  |                  H    *------------------->*
*----------------------*    |                    |          12
           |                |           *----------------------*
           |                |           |   CV = - CV*ITM      |
           |          08    |           |                      |
*----------------------*    |           |   CH = CH*ITM        |
|       C4FT =         |    |           |                      |
|   C4*C0*FT4*PS4      |    |           |    FVVR(I) =         |
*----------------------*    |           |    REAL(CV)          |
           |                |           |                      |
           |                |           |    FVVI(I) = -       |
           |          09    |           |     AIMAG(CV)        |
*----------------------*    |           *----------------------*
| CV = CV + (CV1 +     |    |                      |
|   CV3 - C2FT -       |    |                      |
|     C4FT)*Q1         |    |                      |          13
|                      |    |           *----------------------*
| CH = CH + (CH1 +     |    |           |     FHHR(I) =        |
|   CH3 + C2FT +       |    |           |     REAL(CH)         |
|     C4FT)*Q1         |    |           |                      |
*----------------------*    |           |     FHHI(I) = -      |
           |                |           |      AIMAG(CH)       |
 41.22*-->|                 |           *----------------------*
   EOO    |  NOTE   10      |                      |
```

2


08
C4FT =
C4*C0*FT4*PS4
09
CV = CV + (CV1 +
CV3 - C2FT -
C4FT)*Q1
CH = CH + (CH1 +
CH3 + C2FT +
C4FT)*Q1
41.22*-->
800
NOTE 10
CONTINUE
41.07*-->
801
NOTE 11
CONTINUE
CV = - CV*ITM
CH = CH*ITM
EVVR(I) =
REAL(CV)
EVVI(I) = -
AIMAG(CV)
13
EHHR(I) =
REAL(CH)
EHHI(I) = -
AIMAG(CH)
50
14
END OF DO
LOOP?
NO
40
11
YES
15
EXIT

3

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
            COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, TO
            COMPLEX    PS1, PS2, PS4,PS1P,PS2P,PS4P,  C1,  C2,
              C4,  C3,  C9, C11, FT1,       FT2, FT4, FT5, FT6, FT7, FT8,
             CT1, CT2, CT4, CT6, CT8, C1T, C2T, C2FT, C4FT
            COMPLEX    CV1, CH1, CV2, CH2, CV3, CH3, CV4, CH4, CV5, CH5,
             CV6, CH6, CV7, CH7, CV8, CH8, CV9, CH9,CV10,CH10,CV11,CH11,
            CV12,CH12,  CV,  CH, CVA, CHA
            REAL N1,N2,NC,ITM
            DIMENSION  EVVR(1), EVVI(1), EHHR(1), EHHI(1)
  5001      FORMAT(9F7.2,I2)
  5010      FORMAT (  29H1  PROGRAM INPUT PARAMETERS   ,/,
                      17H  ASPECT ANGLE = , F9.4 )
  6001      FORMAT( //, 7H  H1 = ,F8.4 ,7H  A1 = ,F8.4,/  ,7H  H2 = ,F8.4 ,
               7H  A2 = ,F8.4,/  ,7H  H3 = ,F8.4 , 7H  A4 = ,F8.4,/  ,
               7H  H4 = ,F8.4,/  ,7H  H5 = ,F8.4   )
  6005      FORMAT ( ///, 8H  X1D = ,F7.3, 8H  X2D = ,F7.3, / ,
                          8H  X3D = ,F7.3, 8H  X4D = ,F7.3  )
```

CHART TITLE - SUBROUTINE DIFFC(V,H,N,D,D1,E,PHI)

```
                      -----------
                     /  DIFFC   /
                    -----------
                          |
              34.30*-->|
                          |                 01
           *---------------------------*
           |     D2 = 1./(D -          |
           |      COS((PHI +           |
           |       PHI)/N))            |
           |                           |
           |   V = E*(D1 - D2)         |
           |                           |
           |   H = E*(D1 + D2)         |
           *---------------------------*
                          |
                          |
                          | 02
                      ---------
                     *  EXIT   *
                      ---------
```

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
REAL N
```

CHART TITLE - SUBROUTINE BESL(X,B0,B1,B2)

L-122

```
         ------------
       /   BESL    /
        ------------

               *

   * BESSEL FUNCTION
  SUBROUTINE UTILIZING
  POLYNOMIAL
  APPROXIMATIONS
   * COMPUTES J0,J1,OR
  J2 FOR POSITIVE REAL
  ARGUMENTS
   * REFERENCE (HNDBK
  MATH FUNCT BY
  ABRAMOWITZ AND STEGUN
  SECTION 9.4 )

               *
               |           01
    *--------------------*
    |       S = 1.0      |
    *--------------------*
               |
               |
               |
               *    02                          *------------------->*
             *   *                              |          9         |           12
           *       *                            |         *--------------------*
  FALSE  *           *                          |         |     X2 = 3./X      |
*-----*   X .LT. 0.    *                        |         |                    |
|      *           *                            |         | F0 = .79788456 +   |
|        *       *                              |         |   X2*(-.77E-6 +    |
|          *   *                                |         | X2*(-.00552740 +   |
|            *                                  |         |   X2*(-.9512E-4 +  |
|            |TRUE                              |         |   X2*(.00137237 +  |
|            |                                  |         | X2*(-.72805E-3 +   |
|            |                                  |         | X2*0.14476E-3)))) |
|            |                                  |         |         )          |
|            |                                  |         *--------------------*
|            |                                  |                    |
|            |           03                     |                    |
|   *--------------------*                      |                    |           13
|   |     S = - 1.       |                      |         *--------------------*
|   *--------------------*                      |         |     T0 = X -       |
|            |                                  |         |    .78539816 +     |
*----------->|                                  |         | X2*(-.04166397 +   |
             |           04                     |         |   X2*(-.39545E-4 + |
    *--------------------*                      |         |   X2*(.00262573 +  |
    |     X = ABS(X)     |                      |         | X2*(-.00054125 +   |
    *--------------------*                      |         | X2*(-.00029333 +   |
             |                                  |         | X2*0.00013558)))) |
             |                                  |         |         )          |
             *     05                           |         *--------------------*
```

2

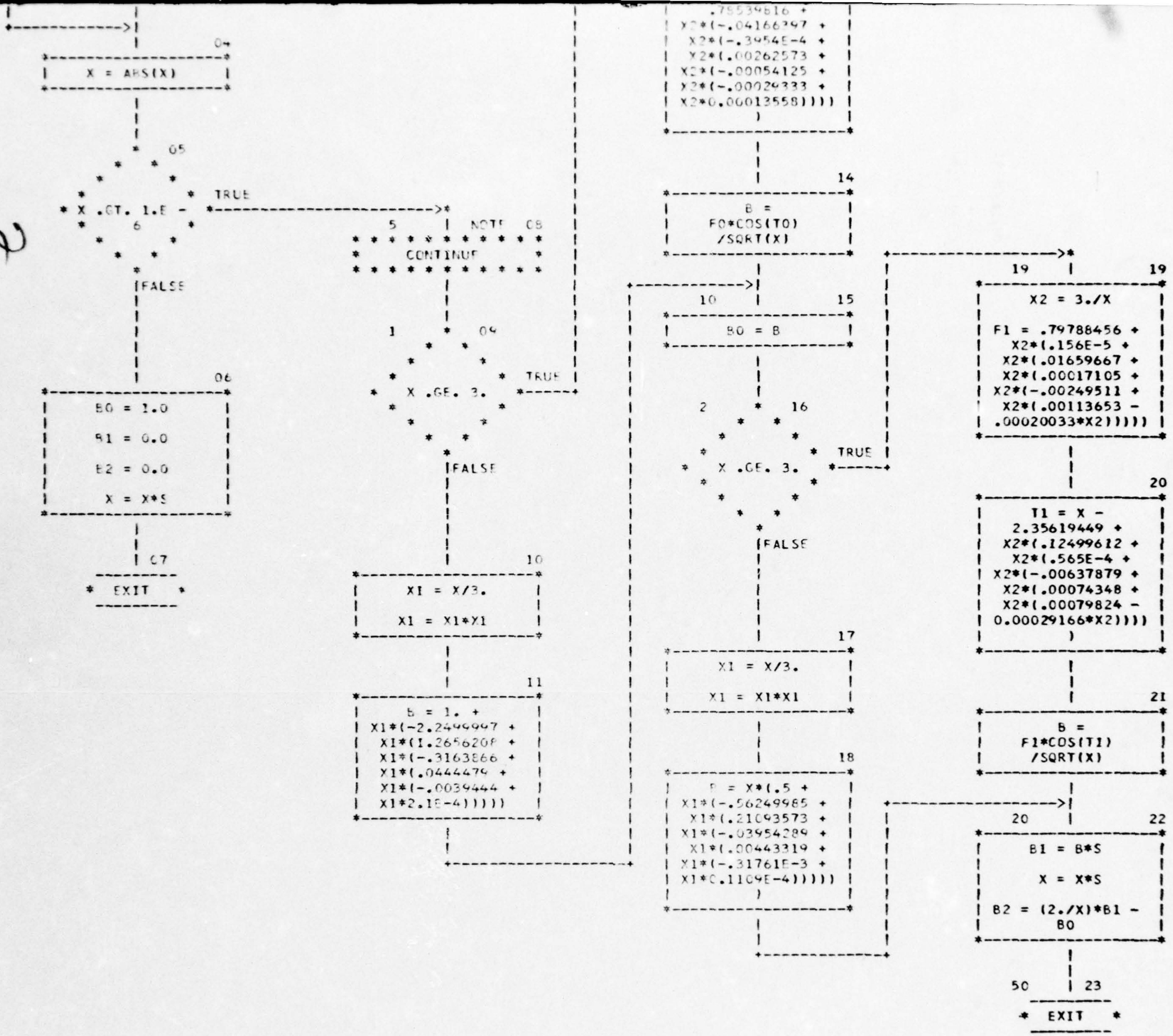
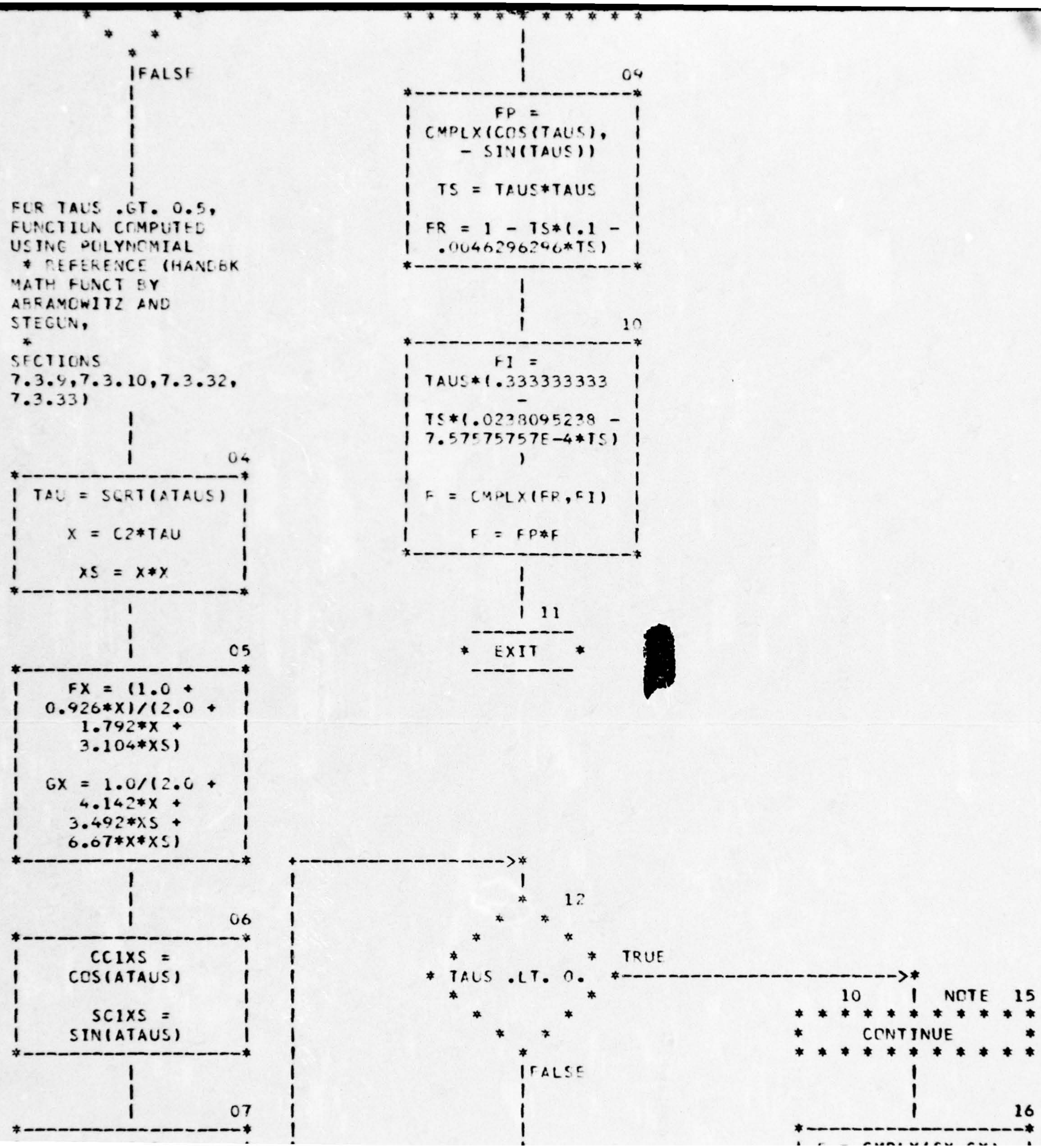

04
X = ABS(X)
05
X .GT. 1.E - 6
TRUE
FALSE
06
B0 = 1.0
B1 = 0.0
B2 = 0.0
X = X*S
07
EXIT
5
NOTE 08
CONTINUE
1
09
X .GE. 3.
TRUE
FALSE
10
X1 = X/3.
X1 = X1*X1
11
B = 1. +
X1*(-2.2499997 +
X1*(1.2656208 +
X1*(-.3163866 +
X1*(.0444479 +
X1*(-.0039444 +
X1*2.1E-4)))))
.79539816 +
X2*(-.04166397 +
X2*(-.39545E-4 +
X2*(.00262573 +
X2*(-.00054125 +
X2*(-.00029333 +
X2*0.00013558))))
)
14
B =
F0*COS(T0)
/SQRT(X)
10
15
B0 = B
2
16
X .GE. 3.
TRUE
FALSE
17
X1 = X/3.
X1 = X1*X1
18
B = X*(.5 +
X1*(-.56249985 +
X1*(.21093573 +
X1*(-.03954289 +
X1*(.00443319 +
X1*(-.31761E-3 +
X1*0.1109E-4)))))
)
19
19
X2 = 3./X
F1 = .79788456 +
X2*(.156E-5 +
X2*(.01659667 +
X2*(.00017105 +
X2*(-.00249511 +
X2*(.00113653 -
.00020033*X2)))))
20
T1 = X -
2.35619449 +
X2*(.12499612 +
X2*(.565E-4 +
X2*(-.00637879 +
X2*(.00074348 +
X2*(.00079824 -
0.00029166*X2))))
)
21
B =
F1*COS(T1)
/SQRT(X)
20
22
B1 = B*S
X = X*S
B2 = (2./X)*B1 -
B0
50
23
EXIT

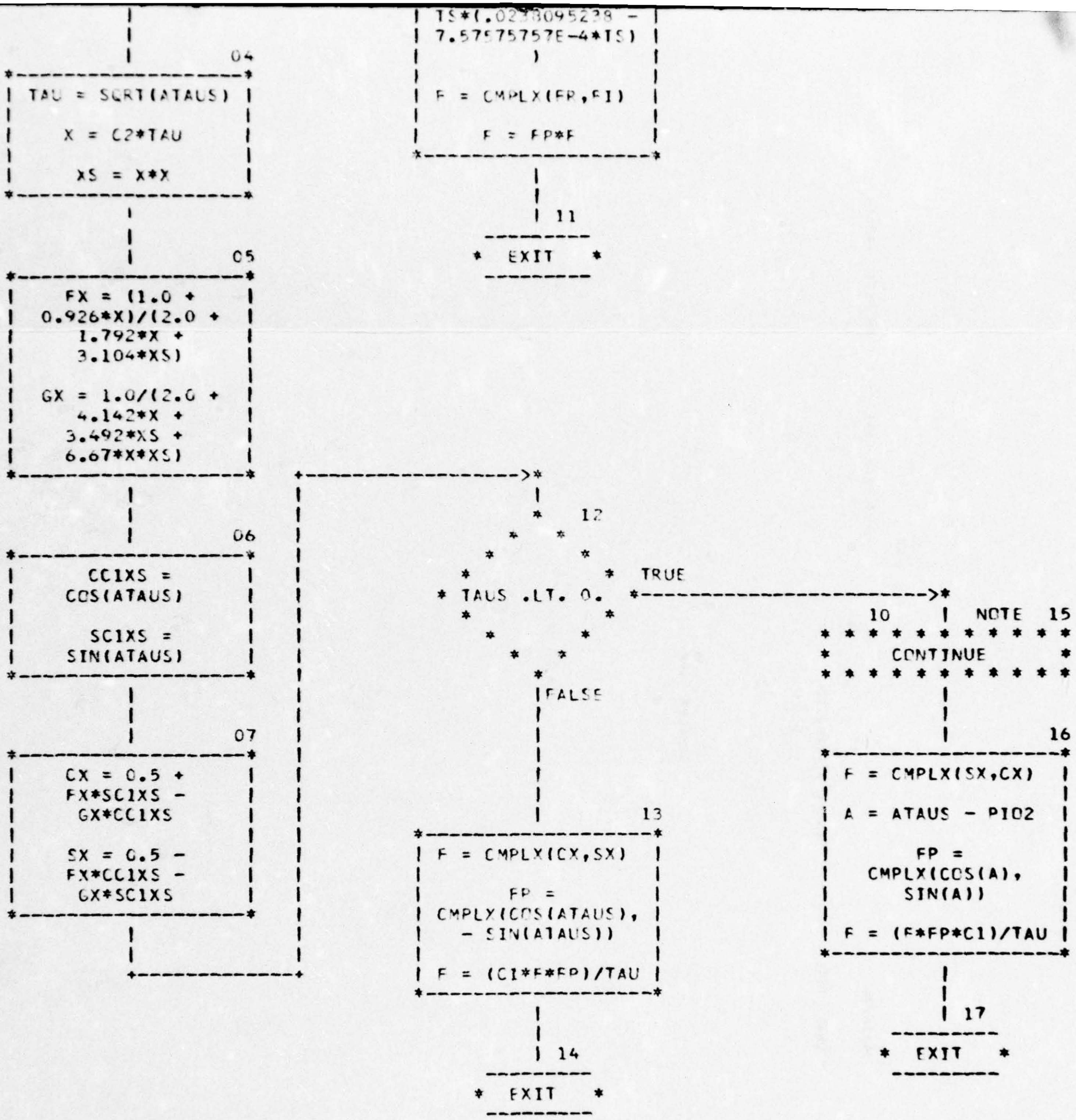
CHART TITLE - SUBROUTINE FTG(TAUS,F)

```
              -----------
            /   FTG     /
             -----------

         40.32*-->*

        COMPUTES FTAU WHERE
        FTAU
        =(EXP(-J*TAU**2)
        /2*TAU)*SQRT(PI/2.)*
        (C2(TAU**2) +
        J*S2(TAU**2))

                   *
                   |        02
        *--------------------*
        |      PI =          |
        | 3.14159265358979   |
        |                    |
        |   PIO2 = PI/2.     |
        |                    |
        | C1 = SQRT(PI/2.)   |
        |                    |
        |    C2 = 1./C1      |
        |                    |
        | ATAUS = ABS(TAUS)  |                 FOR TAUS .LE. 0.5,
        *--------------------*                 FUNCTION IS EXPANDED
                   |                           IN SERIES AND FIRST
                   |                           FEW
                   |                           TERMS INTEGRATED TERM
                   |                           BY TERM TO OBTAIN
                   *   03                      RESULT
                 *   *
               *       *          +------------------->*
             *           *  TRUE  |            20      |  NOTE  08
          *   ATAUS .LE.   *------+          * * * * * * * * * * *
             *   0.5     *                   *      CONTINUE     *
               *       *                     * * * * * * * * * * *
                 *   *                                 |
                   *                                   |
                   |FALSE                              |        09
                   |                         *-------------------*
                   |                         |       FP =        |
                                             | CMPLX(COS(TAUS),  |
```

2



```
FALSE

FOR TAUS .GT. 0.5,
FUNCTION COMPUTED
USING POLYNOMIAL
 * REFERENCE (HANDBK
MATH FUNCT BY
ABRAMOWITZ AND
STEGUN,
 *
SECTIONS
7.3.9,7.3.10,7.3.32,
7.3.33)

04
TAU = SQRT(ATAUS)
X = C2*TAU
XS = X*X

05
FX = (1.0 +
0.926*X)/(2.0 +
1.792*X +
3.104*XS)
GX = 1.0/(2.0 +
4.142*X +
3.492*XS +
6.67*X*XS)

06
CCIXS =
COS(ATAUS)
SCIXS =
SIN(ATAUS)

07

09
FP =
CMPLX(COS(TAUS),
 - SIN(TAUS))
TS = TAUS*TAUS
FR = 1 - TS*(.1 -
.0046296296*TS)

10
FI =
TAUS*(.333333333
 -
TS*(.0238095238 -
7.57575757E-4*TS)
 )
F = CMPLX(FR,FI)
F = FP*F

11
EXIT

12
TAUS .LT. 0.
TRUE
FALSE

10    NOTE 15
CONTINUE

16
```



```
04
TAU = SCRT(ATAUS)
X = C2*TAU
XS = X*X

05
FX = (1.0 +
0.926*X)/(2.0 +
1.792*X +
3.104*XS)
GX = 1.0/(2.0 +
4.142*X +
3.492*XS +
6.67*X*XS)

06
CC1XS =
COS(ATAUS)
SC1XS =
SIN(ATAUS)

07
CX = 0.5 +
FX*SC1XS -
GX*CC1XS
SX = 0.5 -
FX*CC1XS -
GX*SC1XS

TS*(.0238095238 -
7.57575757E-4*TS)
)
F = CMPLX(FR,FI)
F = FP*F

11
EXIT

12
TAUS .LT. 0.
TRUE
FALSE

13
F = CMPLX(CX,SX)
FP =
CMPLX(COS(ATAUS),
- SIN(ATAUS))
F = (CI*F*FP)/TAU

14
EXIT

10 NOTE 15
CONTINUE

16
F = CMPLX(SX,CX)
A = ATAUS - PIO2
FP =
CMPLX(COS(A),
SIN(A))
F = (F*FP*C1)/TAU

17
EXIT
```

3

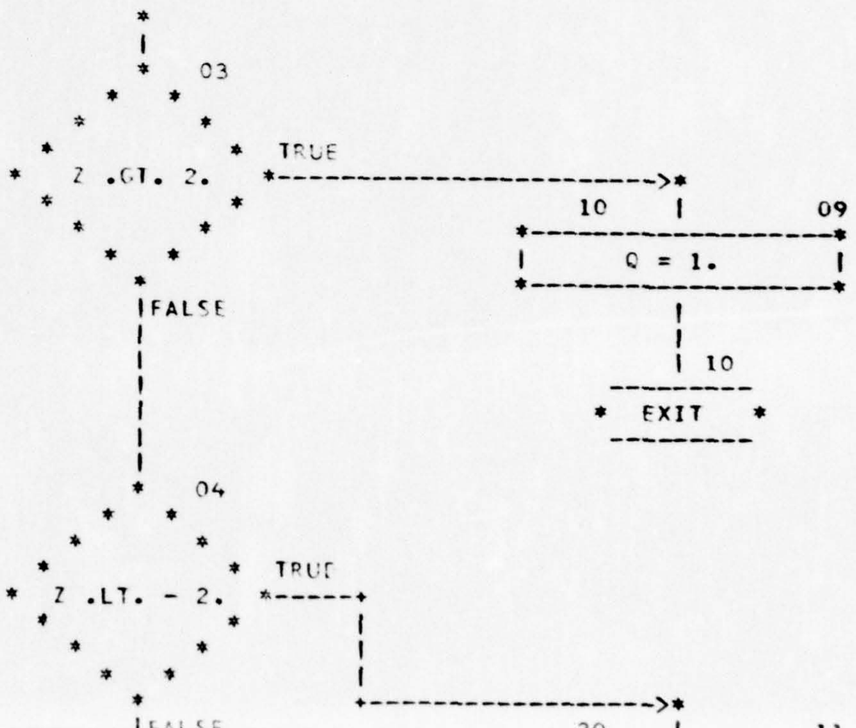
L-123

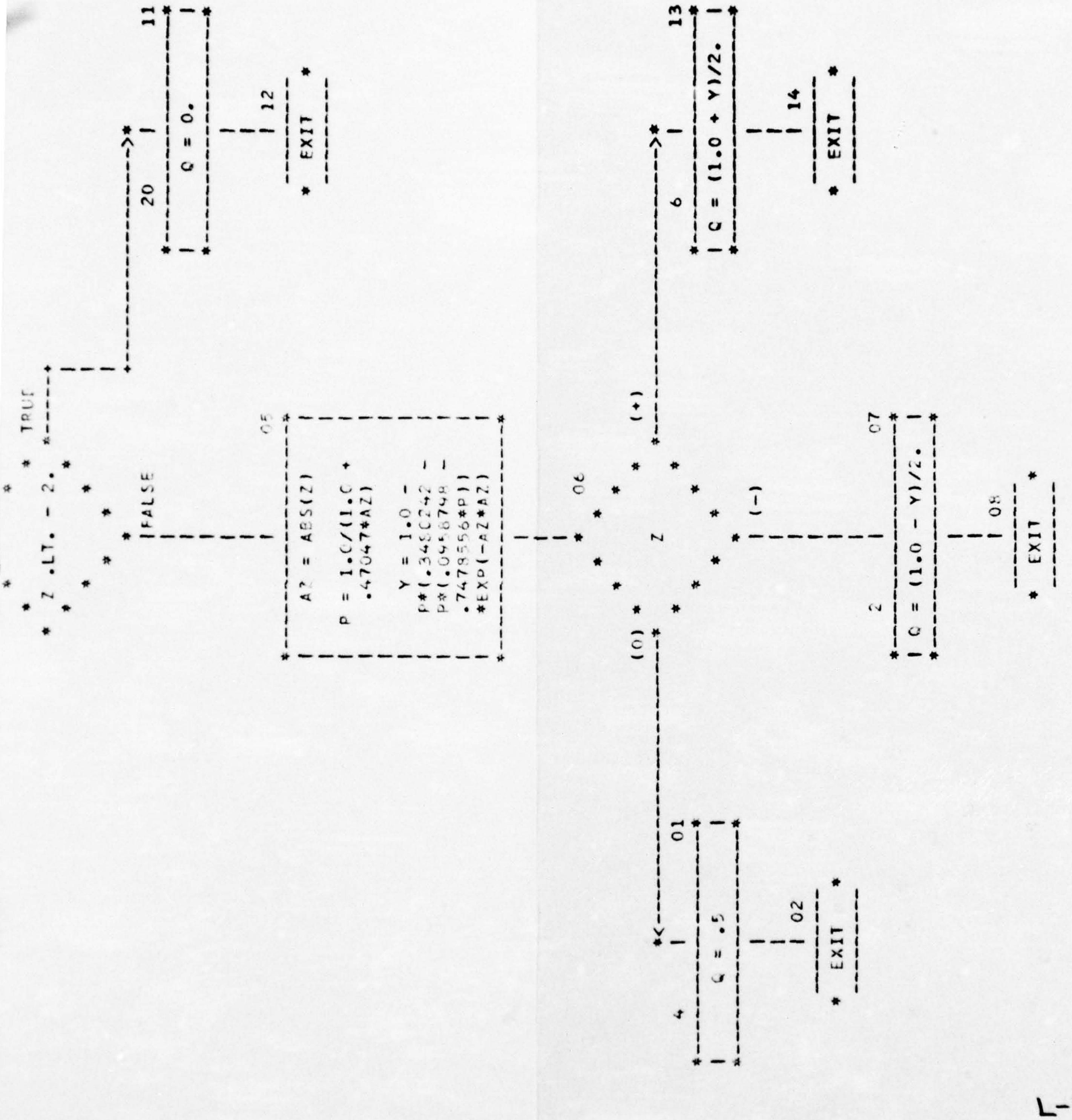
CHART TITLE - NON-PROCEDURAL STATEMENTS

```
COMPLEX F,FP
```

CHART TITLE - FUNCTION Q(Z)

```
Q(Z) = 0.5*(1 +
ERF(Z))
 * ERF(Z) IS
EVALUATED USING A
RATIONAL POLYNOMIAL
APPROXIMATION
 * REFERENCE (HANDBK
MATH FUNCT BY
ABRAMOWITZ AND
STEGUN,
 *           SECTION
7.1.26)
```




Z .LT. - 2.
TRUE
FALSE
20
11
Q = 0.
12
EXIT
05
AZ = ABS(Z)
P = 1.0/(1.0 + .47047*AZ)
Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P)) *EXP(-AZ*AZ)
06
Z
(+)
(-)
(0)
6
13
Q = (1.0 + Y)/2.
14
EXIT
2
07
Q = (1.0 - Y)/2.
08
EXIT
4
01
Q = .5
02
EXIT

L-125

2

```
      SUBROUTINE TARGET  (EVVR,EVVI,EHHR,EHHI )                             RCS8 001
C     **   RESPONSE OF TARGET ST-2                             **           RCS8 002
C     **   COMPUTED UTILIZING THE BUCK-CHIMISEV EQUATIONS   **              RCS8 003
C                                                                           RCS8 004
      COMMON MOVER, M, NMIN. NMAX, DF, FC, PW, TO                           RCS8 005
C     NMIN = MINIMUM FREQUENCY SAMPLE                                       RCS8 006
C     NMAX = MAXIMUM FREQUENCY SAMPLE                                       RCS8 007
C     DF   = FREQUENCY INCREMENT IN MHZ                                     RCS8 008
C     FC   = CARRIER FREQUENCY IN GHZ                                       RCS8 009
      COMPLEX    PS1, PS2, PS4,PS1P,PS2P,PS4P,  C1,  C2,                    RCS8 010
     A C4,  C3,  C9, C11, FT1,       FT2, FT4, FT5, FT6, FT7, FT8,          RCS8 011
     B CT1, CT2, CT4, CT6, CT8, C1T, C3T, C2FT, C4FT                       . RCS8 012
      COMPLEX    CV1, CH1, CV2, CH2, CV3, CH3, CV4, CH4, CV5, CH5,          RCS8 013
     A CV6, CH6, CV7, CH7, CV8, CH8, CV9, CH9,CV10,CH10,CV11,CH11,          RCS8 014
     BCV12,CH12,  CV,  CH, CVA, CHA                                         RCS8 015
      REAL N1,N2,NC,ITM                                                     RCS8 016
      DIMENSION  EVVR(1), EVVI(1), EHHR(1), EHHI(1)                         RCS8 017
C                                                                           RCS8 018
C                                                                           RCS8 019
C                                                                           RCS8 020
C     PROGRAM CONSTANTS                                                     RCS8 021
C                                                                           RCS8 022
      PI = 3.14159265358979                                                 RCS8 023
      PI2 = PI+PI                                                           RCS8 024
      PIO2 = PI/2.                                                          RCS8 025
      HPI  = .5*PI                                                          RCS8 026
```

```
                                                                                 RCS8 023
 1446            PI2 = PI+PI                                                     RCS8 024
 1447            PIO2 = PI/2.                                                    RCS8 025
 1448            HPI  = .5*PI                                                    RCS8 026
 1449            SPI  = SQRT(PI)                                                 RCS8 027
```

2

```
04/26/76     INPUT LISTING                          AUTOFLOW CHART SET - FWD/SCL  RADSIM

 CARD NO          ****                          CONTENTS                             ****

 1450            RTD = 180./PI                                                   RCS8 028
 1451            DTR = PI/180.                                                   RCS8 029
 1452            SHPI = SQRT(HPI)                                                RCS8 030
 1453            SIHPI = 1./SHPI                                                 RCS8 031
 1454            ITM =0.0254                                                     RCS8 032
 1455            SMIN = 1.E-4                                                    RCS8 033
 1456            SMDB = -80.                                                     RCS8 034
 1457            C = 11.80285078                                                 RCS8 035
 1458      C                                                                     RCS8 036
 1459            READ(5,5001) ASPECT,A1,A2,A4,H1L,H2L,H3L,H4L,H5L,M1             RCS8 037
 1460       5001 FORMAT(9F7.2,I2)                                                RCS8 039
 1461            WRITE(6,5010) ASPECT                                            RCS8 040
 1462       5010 FORMAT (  24H1  PROGRAM INPUT PARAMETERS   ,/,                  RCS8 041
 1463          1        17H  ASPECT ANGLE = , F9.4 )                             RCS8 042
 1464            WRITE (6,6001) H1L,A1,H2L,A2,H3L,A4,H4L,H5L                     RCS8 043
 1465       6001 FORMAT( //, 7H
      SHPI = SQRT(HPI)                                                RCS8 030
      SIHPI = 1./SHPI                                                 RCS8 031
      ITM =0.0254                                                     RCS8 032
      SMIN = 1.E-4                                                    RCS8 033
      SMDB = -80.                                                     RCS8 034
      C = 11.80285078                                                 RCS8 035
C                                                                     RCS8 036
      READ(5,5001) ASPECT,A1,A2,A4,H1L,H2L,H3L,H4L,H5L,M1             RCS8 037
 5001 FORMAT(9F7.2,I2)                                                RCS8 039
      WRITE(6,5010) ASPECT                                            RCS8 040
 5010 FORMAT (  29H1  PROGRAM INPUT PARAMETERS  ,/,                   RCS8 041
     1        17H  ASPECT ANGLE = , F9.4 )                            RCS8 042
      WRITE (6,6001) H1L,A1,H2L,A2,H3L,A4,H4L,H5L                     RCS8 043
 6001 FORMAT( //, 7H  H1 = ,F8.4 ,7H  A1 = ,F8.4,/  ,7H  H2 = ,F8.4 , RCS8 044
     A  7H  A2 = ,F8.4,/  ,7H  H3 = ,F8.4 , 7H  A4 = ,F8.4,/  ,       RCS8 045
     B  7H  H4 = ,F8.4,/  ,7H  H5 = ,F8.4  )                          RCS8 046
C                                                                     RCS8 047
      TH = ASPECT*DTR                                                 RCS8 048
C                                                                     RCS8 049
      H12L = H1L + H2L                                                RCS8 050
      H125L = H1L + H2L + H5L                                         RCS8 051
      H23L = H2L + H3L                                                RCS8 052
      H234L = H2L + H3L + H4L                                         RCS8 053
      A21 = A2-A1                                                     RCS8 054
      A42 = A4-A2                                                     RCS8 055
C                                                                     RCS8 056
```

L-125a

3

/ L-125b

```
      X1 = ATAN( A21/H1L)                                               RCS8 057
      X2 = X1                                                           RCS8 058
      X3 = ATAN( A42/(H2L+H23L))                                        RCS8 059
      X4 = ATAN( A21/(H1L+H5L))                                         RCS8 060
C                                                                       RCS8 061
      PIMX3 = PI-X3                                                     RCS8 062
      PIMX4 = PI-X4                                                     RCS8 063
C                                                                       RCS8 064
      X1D = X1*RTD                                                      RCS8 065
      X2D = X2*RTD                                                      RCS8 066
      X3D = X3*RTD                                                      RCS8 067
      X4D = X4*RTD                                                      RCS8 068
C                                                                       RCS8 069
      WRITE (6,6005) X1D,X2D,X3D,X4D                                    RCS8 070
 6005 FORMAT ( ///, 8H   X1D = ,F7.3, 8H   X2D = ,F7.3, / ,             RCS8 071
     A              8H   X3D = ,F7.3, 8H   X4D = ,F7.3  )               RCS8 072
C                                                                       RCS8 073
      N1 = 1.-X1/PI                                                     RCS8 074
      N2 = 1.+X1/PI                                                     RCS8 075
      NC = 1.5                                                          RCS8 076
C                                                                       RCS8 077
      D1 = COS(PI/N1)                                                   RCS8 078
      D2 = COS(PI/N2)                                                   RCS8 079
      DC = -.5                                                          RCS8 080
C                                                                       RCS8 081
      D11 = 1./(D1-1.)                                                  RCS8 082
```

```
    1501              DC = -.5                                                        RCS8 080
    1502        C                                                                     RCS8 081
    1503              D11 = 1./(D1-1.)                                                RCS8 082
    1504              D21 = 1./(D2-1.)                                                RCS8 083
    1505              DC1 = -2./3.                                                    RCS8 084
    1506        C                                                                     RCS8 085
    1507              E1 = SIN(PI/N1)/N1                                              RCS8 086
```

```
    1508              E2 = SIN(PI/N2)/N2                                              RCS8 087
    1509              FC = SIN(PI/NC)/NC                                              RCS8 088
    1510        C                                                                     RCS8 089
    1511              STH = SIN(TH)                                                   RCS8 090
    1512              CTH = COS(TH)                                                   RCS8 091
    1513        C                                                                     RCS8 092
    1514              X0 =(2.*PI)/C                                                   RCS8 093
    1515              AX0=(X0/1000.)                                                  RCS8 094
    1516              XAK2 = X0*FC                                                    RCS8 095
    1517              AK2A1 =XAK2*A1                                                  RCS8 096
    1518              AK2A2 =XAK2*A2                                                  RCS8 097
    1519              AK2A4 =XAK2*A4                                                  RCS8 098
    1520        C                                                                     RCS8 099
```

```
      X0 =(2.*PI)/C                                               RCS8 093
      AX0=(X0/1000.)                                              RCS8 094
      XAK2 = X0*FC                                                RCS8 095
      AK2A1 =XAK2*A1                                              RCS8 096
      AK2A2 =XAK2*A2                                              RCS8 097
      AK2A4 =XAK2*A4                                              RCS8 098
C                                                                 RCS8 099
      XPT  = X1+TH                                                RCS8 100
      C0 = TAN(XPT)*0.5                                           RCS8 101
      TSP = COS(XPT)/SIN(X1)                                      RCS8 102
      IF ( TH .GT. PIO2) GO TO 20                                 RCS8 103
      XMT = X1-TH                                                 RCS8 104
      CK0 = 0.5*TAN(XMT)                                          RCS8 105
      TSM = COS(XMT)/SIN(X1)                                      RCS8 106
      Z6 = AK2A2*(X4-TH)                                          RCS8 107
      Z8 = AK2A4*(X3-TH)                                          RCS8 108
      Q8= Q(Z8)                                                   RCS8 109
      Q6= Q(Z6)                                                   RCS8 110
C     DIFFC COMPUTES  C(N)-/+B(N,PHI) TERMS RETURNED AS VX,HX     RCS8 111
      CALL DIFFC( V12, H12,NC,DC,DC1,EC, PIO2-TH )                RCS8 112
      CALL DIFFC ( V11,H11 ,NC,DC,DC1,EC, PIO2+TH )               RCS8 113
      CALL DIFFC ( V1,H1, N1,D1,D11,E1,TH )                       RCS8 114
```

3

```
      CALL DIFFC ( V2,H2, N2,D2,D21,E2, XPT   )                         RCS8 115
      CALL DIFFC ( V9,H9 ,NC,DC,DC1,EC, TH )                            RCS8 116
      CALL DIFFC ( V6,H6, N2,D2,D21,E2,XMT )                            RCS8 117
      GO TO 30                                                          RCS8 118
   20 CONTINUE                                                          RCS8 119
      PIMTH = PI-TH                                                     RCS8 120
      Z2 = AK2A2*(PIMX3-TH)                                             RCS8 121
      PIMXPT = PI-X1-TH                                                 RCS8 122
      Z1 =  AK2A1*PIMXPT*3.                                             RCS8 123
      Z11= AK2A1*(PIMX4-TH)                                             RCS8 124
      Q11= Q(Z11)                                                       RCS8 125
      Q2 = Q(Z2)                                                        RCS8 126
      Q1 = Q(Z1)                                                        RCS8 127
C                                                                       RCS8 128
      CALL DIFFC ( V9,H9,NC,DC,DC1,EC, TH )                             RCS8 129
      CALL DIFFC ( V10,H10,NC,DC,DC1,EC, TH-PIO2 )                      RCS8 130
      CALL DIFFC ( V4,H4 ,N2,D2,D21,E2, PIMTH )                         RCS8 131
      CALL DIFFC ( V1,H1, N1,D1,D11,E1,PIMXPT)                          RCS8 132
      CALL DIFFC(V11,H11,NC,DC,DC1,EC, PIMTH )                          RCS8 133
   30 CONTINUE                                                          RCS8 134
      DO 50  I= NMIN,NMAX                                               RCS8 135
      XI = I-1                                                          RCS8 136
      AK =AXO*XI*DF                                                     RCS8 137
      AK2 = 2.0*AK                                                      RCS8 138
      AK2A1 = AK2*A1                                                    RCS8 139
```

```
     1559             AK2 = 2.0*AK                                                  RCS8 138
     1560             AK2A1 = AK2*A1                                                RCS8 139
     1561             AK2A2 = AK2*A2                                                RCS8 140
     1562             AK2A4 = AK2*A4                                                RCS8 141
2    1563      C                                                                    RCS8 142
     1564             XC1=AK2A1*STH                                                 RCS8 143
     1565             XC2=AK2A2*STH                                                 RCS8 144
```

```
04/26/76       INPUT LISTING                          AUTOFLOW CHART SET - FWD/SCL  RADSIM

 CARD NO       ****                           CONTENTS                                  ****

     1566             XC4=AK2A4*STH                                                 RCS8 145
     1567      C                                                                    RCS8 146
     1568             CALL BESL(XC1,B10,B11,B12)                                    RCS8 147
     1569             CALL BESL(XC2,B20,B21,B22)                                    RCS8 148
     1570             CALL BESL(XC4,B40,B41,B42)                                    RCS8 149
     1571      C                                                                    RCS8 150
     1572      C      PHASE AND AMPLITUDE TERMS FROM BESSEL FUNCTIONS               RCS8 151
     1573             PS1 = CMPLX( B10,-B11)                                        RCS8 152
     1574             PS2 = CMPLX( B20,-B21)                                        RCS8 153
     1575             PS4 = CMPLX( B40,-B41)                                        RCS8 154
     1576             PS1P = CONJG(PS1)                                             RCS8 155
     1577             PS2P = CONJG(PS2)                                             RCS8 156
     1578             PS4P = CONJG(PS4)                                             RCS8 157
```

```
C                                                                 RCS8 146
      CALL BESL(XC1,B10,B11,B12)                                  RCS8 147
      CALL BESL(XC2,B20,B21,B22)                                  RCS8 148
      CALL BESL(XC4,B40,B41,B42)                                  RCS8 149
C                                                                 RCS8 150
C     PHASE AND AMPLITUDE TERMS FROM BESSEL FUNCTIONS             RCS8 151
      PS1 = CMPLX( B10,-B11)                                      RCS8 152
      PS2 = CMPLX( B20,-B21)                                      RCS8 153
      PS4 = CMPLX( B40,-B41)                                      RCS8 154
      PS1P = CONJG(PS1)                                           RCS8 155
      PS2P = CONJG(PS2)                                           RCS8 156
      PS4P = CONJG(PS4)                                           RCS8 157
C                                                                 RCS8 158
C     PHASE TERM USING LENGTH ALONG AXIS                          RCS8 159
      PC1 = -AK2*H12L*CTH                                         RCS8 160
      PC2 = -AK2*H2L*CTH                                          RCS8 161
      PC4 = AK2*H23L*CTH                                          RCS8 162
      PC9 = AK2*H234L*CTH                                         RCS8 163
      PC11=-AK2*H125L*CTH                                         RCS8 164
C                                                                 RCS8 165
      C1    = CMPLX(COS(PC1),SIN(PC1))                            RCS8 166
      C2    = CMPLX(COS(PC2),SIN(PC2))                            RCS8 167
      C4    = CMPLX(COS(PC4),SIN(PC4))                            RCS8 168
      C3    = CONJG(  C2)                                         RCS8 169
      C9    = CMPLX(COS(PC9),SIN(PC9))                            RCS8 170
      C11   =  CMPLX(COS(PC11),SIN(PC11))                         RCS8 171
C                                                                 RCS8 172
```

3

L-125C

L-125d

```
      C1 = A1*SPI*C1                                                    RCS8 173
      C3 = A2*SPI*C3                                                    RCS8 174
      C2 = A2*SPI*C2                                                    RCS8 175
      C4 = A4*SPI*C4                                                    RCS8 176
      C9 = A4*SPI*C9                                                    RCS8 177
      C11 = A1*SPI*C11                                                  RCS8 178
C                                                                       RCS8 179
C     CAUSTIC CORRECTION TERMS                                          RCS8 180
      CC11 = EC*DC1*(B10+B12)                                           RCS8 181
      CC1  = E1*D11*(B10+B12)                                           RCS8 182
      CC2  = E2*D21*(B20+B22)                                           RCS8 183
      CC3  = E1*D11*(B20+B22)                                           RCS8 184
      CC4  = E2*D21*(B40+B42)                                           RCS8 185
      CC9  = EC*DC1*(B40+B42)                                           RCS8 186
C                                                                       RCS8 187
      TSPP = TSP*AK2                                                    RCS8 188
      TS1= A1*TSPP                                                      RCS8 189
      TS2= A2*TSPP                                                      RCS8 190
      TS4= A4*TSPP                                                      RCS8 191
C                                                                       RCS8 192
      IF ( TH .GT. PIO2 ) GO TO 450                                     RCS8 193
C                                                                       RCS8 194
C                                                                       RCS8 195
      CV12 = C11*(V12*PSIP - CC11)                                      RCS8 196
      CH12 = C11*(H12*PSIP - CC11)                                      RCS8 197
```

```
   1617              CV12 = C11*(V12*PS1P - CC11)                        RCS8 196
   1618              CH12 = C11*(H12*PS1P - CC11)                        RCS8 197
   1619              CV11 = C11*(V11*PS1 - CC11)                         RCS8 198
   1620              CH11 = C11*(H11*PS1 - CC11)                         RCS8 199
   1621        C                                                         RCS8 200
   1622              CALL FTG (TS1,FT1)                                  RCS8 201
   1623              CT1 = C0*FT1                                        RCS8 202




 04/26/76     INPUT LISTING             AUTOFLOW CHART SET - FWD/SCL  RADSIM

  CARD NO      ****                    CONTENTS                          ****

   1624              CV1 = C1*((V1+CT1)*PS1 - CC1)                       RCS8 203
   1625              CH1 = C1*((H1-CT1)*PS1 - CC1)                       RCS8 204
   1626        C                                                         RCS8 205
   1627              CALL FTG (TS2,FT2)                                  RCS8 206
   1628              CT2 = C0*FT2                                        RCS8 207
   1629              CV3 = C3*((V1+CT2)*PS2 - CC3)                       RCS8 208
   1630              CH3 = C3*((H1-CT2)*PS2 - CC3)                       RCS8 209
   1631        C                                                         RCS8 210
   1632              CV2 = C2*((V2-CT2)*PS2 - CC2)                       RCS8 211
   1633              CH2 = C2*((H2+CT2)*PS2 - CC2)                       RCS8 212
   1634        C                                                         RCS8 213
   1635              CALL FTG (TS4,FT4)                                  RCS8 214
   1636              CT4 = C0*FT4                                        RCS8 215
```

```
      CH3 = C3*((H1-CT2)*PS2 - CC3)                                     RCS8 209
C                                                                       RCS8 210
      CV2 = C2*((V2-CT2)*PS2 - CC2)                                     RCS8 211
      CH2 = C2*((H2+CT2)*PS2 - CC2)                                     RCS8 212
C                                                                       RCS8 213
      CALL FTC (TS4,FT4)                                                RCS8 214
      CT4 = C0*FT4                                                      RCS8 215
      CV4 = C4*((V2-CT4)*PS4 - CC4)                                     RCS8 216
      CH4 = C4*((H2+CT4)*PS4 - CC4)                                     RCS8 217
C                                                                       RCS8 218
      CV9 = C9*(V9*PS4 - CC9)                                           RCS8 219
      CH9 = C9*(H9*PS4 - CC9)                                           RCS8 220
C                                                                       RCS8 221
      CV = CV11+CV12+CV1+CV2+CV3+CV4+CV9                                RCS8 222
      CH = CH11+CH12+CH1+CH2+CH3+CH4+CH9                                RCS8 223
C                                                                       RCS8 224
      IF ( Z6 .LE.-2.)GO TO 801                                         RCS8 225
C                                                                       RCS8 226
      TSMM = TSM*AK2                                                    RCS8 227
      TS5 =TSMM*A1                                                      RCS8 228
      TS6 =TSMM*A2                                                      RCS8 229
      TS8 =TSMM*A4                                                      RCS8 230
```

3

```
C                                                                       RCS8 231
C                                                                       RCS8 232
C                                                                       RCS8 233
      CALL FTG(TS6,FT6)                                                 RCS8 234
      CT6 = CK0*FT6                                                     RCS8 235
      CV6 = C2*((V6-CT6)*PS2P - CC2)*Q6                                 RCS8 236
      CH6 = C2*((H6+CT6)*PS2P - CC2)*Q6                                 RCS8 237
C                                                                       RCS8 238
      CALL FTG(TS5,FT5)                                                 RCS8 239
      FT5 = CK0*(1.-FT5)*PS1P*C1*Q6                                     RCS8 240
      CV5 = -FT5                                                        RCS8 241
      CH5 = +FT5                                                        RCS8 242
C                                                                       RCS8 243
      FT7 = CK0*(1.-FT6)*PS2P*C3*Q8                                     RCS8 244
      CV7 = -FT7                                                        RCS8 245
      CH7 = FT7                                                         RCS8 246
C                                                                       RCS8 247
      CALL FTG(TS8,FT8)                                                 RCS8 248
      CT8 = CK0*FT8                                                     RCS8 249
      CV8 = C4*((V6-CT8)*PS4P - CC4)*Q8                                 RCS8 250
      CH8 = C4*((H6+CT8)*PS4P - CC4)*Q8                                 RCS8 251
C                                                                       RCS8 252
      CVA = CV5 + CV6 + CV7 + CV8                                       RCS8 253
      CHA = CH5 + CH6 + CH7 + CH8                                       RCS8 254
C                                                                       RCS8 255
                                                                        RCS8 256
```

```
  1675            CHA = CH5 + CH6 + CH7 + CH8                              RCS8 254
  1676      C                                                              RCS8 255
  1677            CV = CV+CVA                                              RCS8 256
  1678            CH = CH+CHA                                              RCS8 257
  1679            GO TO 801                                                RCS8 258
  1680      C                                                              RCS8 259
  1681        450 CONTINUE                                                 RCS8 260




04/26/76     INPUT LISTING                    AUTOFLOW CHART SET - FWO/SCL  RADSIM

 CARD NO     ****                          CONTENTS                          ****

  1682      C                                                              RCS8 261
  1683      C     THETA GREATER THAN PI/2                                  RCS8 262
  1684      C                                                              RCS8 263
  1685            CV9 = C9*(V9*PS4 - CC9)                                  RCS8 264
  1686            CH9 = C9*(H9*PS4 - CC9)                                  RCS8 265
  1687      C                                                              RCS8 266
  1688            CV10 = C9*(V10*PS4P - CC9)                               RCS8 267
  1689            CH10 = C9*(H10*PS4P - CC9)                               RCS8 268
  1690      C                                                              RCS8 269
  1691            CV4 = C4*    ( V4*PS4 - CC4)                             RCS8 270
  1692            CH4 = C4*    ( H4*PS4 - CC4)                             RCS8 271
  1693            CV = CV9 + CV10 +CV4                                     RCS8 272
  1694            CH = CH9 + CH10 +CH4                                     RCS8 273
```

```
      CV9 = C9*(V9*PS4 - CC9)                                     RCS8 264
      CH9 = C9*(H9*PS4 - CC9)                                     RCS8 265
C                                                                 RCS8 266
      CV10 = C9*(V10*PS4P - CC9)                                  RCS8 267
      CH10 = C9*(H10*PS4P - CC9)                                  RCS8 268
C                                                                 RCS8 269
      CV4 = C4*    ( V4*PS4 - CC4)                                RCS8 270
      CH4 = C4*    ( H4*PS4 - CC4)                                RCS8 271
      CV = CV9 + CV10 +CV4                                        RCS8 272
      CH = CH9 + CH10 +CH4                                        RCS8 273
C                                                                 RCS8 274
      IF (    Z2      .LE. -2.) GO TO 800                         RCS8 275
C                                                                 RCS8 276
      CV11 = C11* (V11*PS1 - CC11)*Q11                            RCS8 277
      CH11 = C11* (H11*PS1 - CC11)*Q11                            RCS8 278
C                                                                 RCS8 279
      CV2 =    C2* (V4*PS2 - CC2)*Q2                              RCS8 280
      CH2 =    C2* (H4*PS2 - CC2)*Q2                              RCS8 281
C                                                                 RCS8 282
      CV = CV + CV11 + CV2                                        RCS8 283
      CH = CH + CH11 + CH2                                        RCS8 284
      IF (    Z1 .GT.-2.) GO TO 700                               RCS8 285
      HTTH = 0.5*(STH/CTH)                                        RCS8 286
      C1T = C1*HTTH*PS1*Q11                                       RCS8 287
      C3T = C3*HTTH*PS2*Q2                                        RCS8 288
```

3

L-125e

L-125f

```
      CV = CV + C1T + C3T                                   RCS8 289
      CH = CH - C1T - C3T                                   RCS8 290
      GC TC 800                                             RCS8 291
C                                                           RCS8 292
  700 CONTINUE                                              RCS8 293
C                                                           RCS8 294
      CALL FTG (TS1,FT1 )                                   RCS8 295
      CV1 = C1*   ((V1+C0*FT1)*PS1 - CC1 )                  RCS8 296
      CH1 = C1*   ((H1-C0*FT1)*PS1 - CC1 )                  RCS8 297
      CALL FTG (TS2,FT2 )                                   RCS8 298
      CV3 = C3*   ((V1+C0*FT2)*PS2 - CC3 )                  RCS8 299
      CH3 = C3*   ((H1-C0*FT2)*PS2 - CC3 )                  RCS8 300
C                                                           RCS8 301
      C2FT = C2   *C0*FT2*PS2                               RCS8 302
      CALL FTG (TS4,FT4)                                    RCS8 303
      C4FT = C4   *C0*FT4*PS4                               RCS8 304
C                                                           RCS8 305
      CV = CV +(CV1 + CV3 - C2FT - C4FT)*Q1                 RCS8 306
      CH = CH +(CH1 + CH3 + C2FT + C4FT)*Q1                 RCS8 307
C                                                           RCS8 308
C                                                           RCS8 309
  800 CONTINUE                                              RCS8 310
C                                                           RCS8 311
  801 CONTINUE                                              RCS8 312
      CV =-CV*ITM                                           RCS8 313
      CH = CH +(CH1 + CH3 + C2FT + C4FT)*Q1                             RCS8 307
C                                                                       RCS8 308
C                                                                       RCS8 309
  800 CONTINUE                                                          RCS8 310
C                                                                       RCS8 311
  801 CONTINUE                                                          RCS8 312
      CV =-CV*ITM                                                       RCS8 313
      CH = CH*ITM                                                       RCS8 314
      EVVR(I) = REAL(CV)                                                RCS8 315
      EVVI(I) = -AIMAG(CV)                                              RCS8 316
      EHHR(I) = REAL(CH)                                                RCS8 317
      EHHI(I) = -AIMAG(CH)                                              RCS8 318
   50 CONTINUE                                                          RCS8 319
      RETURN                                                            RCS8 320
      END                                                               RCS8 321
```

2

```
      SUBROUTINE DIFFC ( V,H,N,D,D1,E, PHI )                            RCS8 322
      REAL N                                                            RCS8 323
C                                                                       RCS8 324
      D2 = 1./(D-COS((PHI+PHI)/N))                                      RCS8 325
      V  = E*(D1-D2)                                                    RCS8 326
      H  = E*(D1+D2)                                                    RCS8 327
      RETURN                                                            RCS8 328
      END                                                               RCS8 329
```

L-1259

```
      SUBROUTINE BESL ( X, B0, B1, B2 )                                 RCS8 330
C                                                                       RCS8 331
C   * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS     RCS8 332
C   * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS                   RCS8 333
C   * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )RCS8 334
C                                                                       RCS8 335
C                                                                       RCS8 336
      S = 1.0                                                           RCS8 337
      IF ( X.LT. 0. ) S= -1.                                            RCS8 338
      X = ABS(X)                                                        RCS8 339
      IF ( X .GT. 1.E-6 ) GO TO 5                                       RCS8 340
      B0 = 1.0                                                          RCS8 341
      B1 = 0.0                                                          RCS8 342
      B2 = 0.0                                                          RCS8 343
      X = X*S                                                           RCS8 344
      RETURN                                                            RCS8 345
C                                                                       RCS8 346
```

L-125h

```
      SUBROUTINE FTG(TAUS,F)                                            RCS8 384
C                                                                       RCS8 385
C     COMPUTES FTAU WHERE FTAU =(EXP(-J*TAU**2)/2*TAU)*SQRT(PI/2.)*     RCS8 386
C                                   (C2(TAU**2) + J*S2(TAU**2))         RCS8 387
C                                                                       RCS8 388
      COMPLEX F,FP                                                      RCS8 389
      PI = 3.14159265358979                                             RCS8 390
      PIO2 = PI/2.                                                      RCS8 391
      C1 = SQRT(PI/2.)                                                  RCS8 392
      C2 = 1./C1                                                        RCS8 393
      ATAUS = ABS(TAUS)                                                 RCS8 394
      IF (ATAUS .LE. 0.5   )GO TO 20                                    RCS8 395
C                                                                       RCS8 396
C     FOR TAUS .GT. 0.5, FUNCTION COMPUTED USING POLYNOMIAL             RCS8 397
C    * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,           RCS8 398
C    *                SECTIONS  7.3.9,7.3.10,7.3.32,7.3.33)             RCS8 399
      TAU = SQRT(ATAUS)                                                 RCS8 400
      X = C2*TAU                                                        RCS8 401
      XS = X*X                                                          RCS8 402
C                                                                       RCS8 403
      FX = (1.0+0.926*X)/(2.0+1.792*X+3.104*XS)                         RCS8 404
      GX = 1.0/(2.0+4.142*X+3.492*XS+6.67*X*XS)                         RCS8 405
C                                                                       RCS8 406
      CC1XS = COS(ATAUS)                                                RCS8 407
      SC1XS = SIN(ATAUS)                                                RCS8 408
C                                                                       RCS8 409
```

```
                                                                      RCS8 409
      CX = 0.5 + FX*SC1XS - GX*CC1XS                                  RCS8 410
      SX = 0.5 - FX*CC1XS - GX*SC1XS                                  RCS8 411
C                                                                     RCS8 412
      IF (TAUS .LT. 0.) GO TO 10                                      RCS8 413
      F= CMPLX( CX,SX)                                                RCS8 414
      FP = CMPLX(  COS(ATAUS), -SIN(ATAUS) )                          RCS8 415
      F =(C1*F*FP)/TAU                                                RCS8 416
      RETURN                                                          RCS8 417
C                                                                     RCS8 418
   10 CONTINUE                                                        RCS8 419
      F = CMPLX(SX,CX)                                                RCS8 420
      A = ATAUS-PIO2                                                  RCS8 421
      FP = CMPLX( COS(A),SIN(A) )                                     RCS8 422
      F = (F*FP*C1)/TAU                                               RCS8 423
      RETURN                                                          RCS8 424
C                                                                     RCS8 425
C     FOR TAUS .LE. 0.5, FUNCTION IS EXPANDED IN SERIES AND FIRST FEW RCS8 426
C                          TERMS INTEGRATED TERM BY TERM TO OBTAIN RESULT RCS8 427
   20 CONTINUE                                                        RCS8 428
      FP = CMPLX( COS(TAUS),-SIN(TAUS) )                              RCS8 429
      TS = TAUS*TAUS                                                  RCS8 430
      FR = 1 - TS*( .1 - .0046296296*TS)                              RCS8 431
      FI = TAUS*( .333333333 - TS*(.0238095238 - 7.57575757E-4*TS))   RCS8 432
      F = CMPLX( FR,FI )                                              RCS8 433
      F = FP*F                                                        RCS8 434
      RETURN                                                          RCS8 435
      END                                                             RCS8 436
```

L-125j

```
    5 CONTINUE                                                          RCS8 347
C                                                                       RCS8 348
    1 IF ( X .GE. 3.) GO TO 9                                           RCS8 349
      X1 = X/3.                                                         RCS8 350
      X1 = X1*X1                                                        RCS8 351
      B = 1.+ X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479RCS8 352
     1 +      X1*(-.0039444+ X1*2.1E-4 )))) )                           RCS8 353
      GO TO 10                                                          RCS8 354
C                                                                       RCS8 355
    9 X2 = 3./X                                                         RCS8 356
      F0 =  .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*  RCS8 357
     1     (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )             RCS8 358
      T0 = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573  RCS8 359
     1    +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))          RCS8 360
      B = F0*COS(T0)/SQRT(X)                                            RCS8 361
C                                                                       RCS8 362
   10 B0 = B                                                            RCS8 363
C                                                                       RCS8 364
    2 IF ( X .GE. 3. ) GO TO 19                                         RCS8 365
      X1 = X/3.                                                         RCS8 366
      X1 = X1*X1                                                        RCS8 367
      B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*    RCS8 368
     1               (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )    RCS8 369
      GO TO 20                                                          RCS8 370
```

L-125C

3

```
    2 IF ( X .GE. 3. ) GO TO 19                                       RCS8 365
      X1 = X/3.                                                       RCS8 366
      X1 = X1*X1                                                      RCS8 367
      B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*  RCS8 368
     1              (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )   RCS8 369
      GO TO 20                                                        RCS8 370
C                                                                     RCS8 371
   19 X2 = 3./X                                                       RCS8 372
      F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*  RCS8 373
     1     (-.00249511 +X2*(.00113653 -.00020033*X2 )))))             RCS8 374
      T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879 RCS8 375
     1     +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))         RCS8 376
      B = F1*COS(T1)/SQRT(X)                                          RCS8 377
C                                                                     RCS8 378
   20 B1 = B*S                                                        RCS8 379
      X = X*S                                                         RCS8 380
      B2= (2./X)*B1 - B0                                              RCS8 381
   50 RETURN                                                          RCS8 382
      END                                                             RCS8 383
      FUNCTION Q(Z)                                                     RCS8 437
C     Q(Z) = 0.5*(1 + ERF(Z))                                           RCS8 438
C   * ERF(Z) IS EVALUATED USING A RATIONAL POLYNOMIAL APPROXIMATION     RCS8 439
C   * REFFRENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,            RCS8 440
C   *             SECTION 7.1.26)                                       RCS8 441
C                                                                       RCS8 442
      IF ( Z.GT. 2.) GO TO 10                                           RCS8 443
      IF ( Z.LT.-2.) GO TO 20                                           RCS8 444
      AZ = ABS(Z)                                                       RCS8 445
      P = 1.0/(1.0 + .47047*AZ)                                         RCS8 446
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)    RCS8 447
      IF (Z) 2,4,6                                                      RCS8 448
    2 Q = (1.0 - Y)/2.                                                  RCS8 449
      RETURN                                                            RCS8 450
    4 Q = .5                                                            RCS8 451
      RETURN                                                            RCS8 452
    6 Q = (1.0 + Y)/2.                                                  RCS8 453
      RETURN                                                            RCS8 454
   10 Q = 1.                                                            RCS8 455
      RETURN                                                            RCS8 456
   20 Q = 0.                                                            RCS8 457
      RETURN                                                            RCS8 458
      END                                                               RCS8 459
```

L-125K

L.7 STEPPED - CYLINDER

The far-field scattering from a stepped-cylinder target configuration shown in Figure L.7-1 has been formulated using the basic Ruck-Ufimtsev technique and a rough approximation of the multiple reflections of the field components between the cylindrical and flat surfaces (Ref. 8).. The basic scattering formulation is the following:

$$e(\theta)_{\substack{\{V\}\\\{H\}}} = \mp\sqrt{\pi}\,\Big\{\, g(11) + g(12) + g(1) + g(2) + g(3) + g(4)$$
$$+ g(9) + g(5) + g(6) + g(7) + g(8) + g(10) \Big\}$$

where g(m) represents the sum of the fringe wave scattering and physical optics response associated with edge m.

For $0 < \theta < \pi/2$,

$$g(12) = a_1 e^{ip_{11}} \Big\{ JJ_{11+} \Big[C(1.5) \mp B(1.5, \pi/2-\theta) \Big] - C(1.5) JJ_{21} \Big\}$$

$$g(11) = a_1 e^{ip_{11}} \Big\{ JJ_{11-} \Big[C(1.5) \mp B(1.5, \pi/2+\theta) \Big] - C(1.5) JJ_{21} \Big\}$$

$$g(2) = a_2 e^{ip2} \Big\{ JJ_{12-} \Big[C(1.5) \mp B(1.5, \pi/2+\theta) \Big] - C(1.5) JJ_{22} \Big\}$$

$$g(4) = a_4 e^{ip_4} \Big\{ JJ_{14-} \Big[C(1.5) \mp B(1.5, \pi/2+\theta) \Big] - C(1.5) JJ_{24} \Big\}$$

$$g(9) = a_4 e^{ip_9} \Big\{ JJ_{14-} \Big[C(1.5) \mp B(1.5, \theta) \Big] - C(1.5) JJ_{24} \Big\}$$

$$g(1) = \pm\, a_1 e^{ip_2} \Big\{ 0.5\ JJ_{11-} \Big[\tan\theta + \cot\theta \Big] \Big\}$$

$$g(3) = \pm\, a_2 e^{ip_4} \Big\{ 0.5\ JJ_{12-} \Big[\tan\theta + \cot\theta \Big] \Big\}$$

$$g(6) = a_2 e^{ip_2} \Big\{ JJ_{12+} \Big[C(1.5) \mp B(1.5, \pi/2-\theta) \Big]$$
$$- C(1.5) JJ_{22} \Big\} Q_6 IQ_6$$

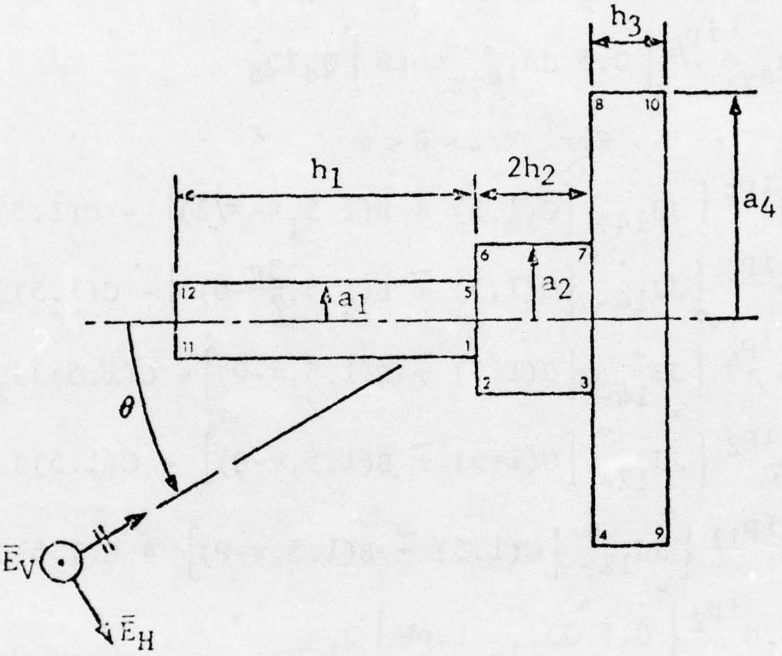


Fig. L.7-1 GEOMETRY OF STEPPED-CYLINDER

$$g(8) = a_4 e^{ip_4} \left\{ JJ_{14+} \left[C(1.5) \mp B(1.5, \pi/2-\theta) \right] - C(1.5) JJ_{24} \right\} Q_8 IQ_8$$

$$g(5) = \pm a_{s_5} e^{ip_2} \left\{ 0.5 \; JJ_{1s_5+} \cot\theta \right\} Q_6 IQ_6$$

$$g(7) = \mp a_{s_7} e^{ip_4} \left\{ 0.5 \; JJ_{1s_7+} \cot\theta \right\} Q_8 IQ_8$$

For $\pi/2 < \theta < \pi$

$$g(10) = a_4 e^{ip_9} \left\{ JJ_{14+} \left[C(1.5) \mp B(1.5, \theta-\pi/2) \right] - C(1.5) JJ_{24} \right\}$$

$$g(9) = a_4 e^{ip_9} \left\{ JJ_{14-} \left[C(1.5) \mp B(1.5, \tfrac{3\pi}{2}-\theta) \right] - C(1.5) JJ_{24} \right\}$$

$$g(4) = a_4 e^{ip_4} \left\{ JJ_{14-} \left[C(1.5) \mp B(1.5, \pi-\theta) \right] - C(1.5) JJ_{24} \right\} Q_4$$

$$g(2) = a_2 e^{ip_2} \left\{ JJ_{12-} \left[C(1.5) \mp B(1.5, \pi-\theta) \right] - C(1.5) JJ_{22} \right\} Q_2$$

$$g(11) = a_1 e^{ip_{11}} \left\{ JJ_{11-} \left[C(1.5) \mp B(1.5, \pi-\theta) \right] - C(1.5) JJ_{21} \right\} Q_{11}$$

$$g(1) = \pm a_1 e^{ip_2} \left\{ 0.5 \; JJ_{11-} \tan\theta \right\} Q_1$$

$$g(3) = \pm a_2 e^{ip_4} \left\{ 0.5 \; JJ_{12-} \tan\theta \right\} Q_3$$

where the upper and lower signs in the previous equations correspond to the case of vertical and horizontal polarization, respectively, and

$$C(n) = \frac{\sin\frac{\pi}{n}}{n} \left[\frac{1}{\cos\frac{\pi}{n} - 1} \right]$$

$$B(n,\emptyset) = \frac{\sin\frac{\pi}{n}}{n} \left[\frac{1}{\cos\frac{\pi}{n} - \cos\frac{2\emptyset}{n}} \right]$$

$$JJ_{1m\mp} = J_o(2ka_m \sin\theta) \mp J_1(2ka_m \sin\theta)$$

$$JJ_{2m} = J_o(2ka_m\sin\theta) + J_2(2ka_m\sin\theta)$$

$$p_{11} = -2k(h_1 + h_2)\cos\theta$$

$$p_2 = -2k\,h_2\cos\theta$$

$$p_4 = 2k\,h_2\cos\theta$$

$$p_5 = 2k(h_2 + h_3)\cos\theta$$

$$Q_{\binom{6}{8}} = Q(2ka_{\binom{2}{4}}(\alpha_{\binom{1}{3}} - \theta)) \quad ; \quad Q_1 = Q(2ka_1(\tfrac{\pi}{2}+\delta-\theta))$$

$$Q_{\binom{11}{2}} = Q(2ka_{\binom{1}{2}}(\pi-\alpha_{\binom{3}{2}}-\theta)) \quad ; \quad Q_3 = Q(2ka_2(\tfrac{\pi}{2}+\delta-\theta))$$

$$\delta = \frac{\pi}{12}$$

$$Q_4 = Q(2ka_4(\pi-\theta))$$

$$IQ_n = \begin{cases} 1 \text{ for } Q_n > 0.5 \\ 0 \text{ otherwise} \end{cases}$$

$$\alpha_1 = \tan^{-1}\left(\frac{a_2-a_1}{h_1}\right)$$

$$\alpha_2 = \tan^{-1}\left(\frac{a_4-a_2}{2h_2}\right)$$

$$\alpha_3 = \tan^{-1}\left(\frac{a_4-a_1}{h_1+2h_2}\right)$$

$$Q(z) = 0.5\left[1 + \text{erf}(z)\right]$$

$$a_{s_5} = a_1 + h_1\tan\theta$$

$$a_{s_7} = \begin{cases} a_2 + 2h_2\tan\theta, \text{ for } \theta < \alpha_1 \\ a_1 + (2h_2+h_1)\tan\theta, \text{ for } \alpha_1 < \theta < \alpha_3 \end{cases}$$

Most of the equations can be obtained in a logical and straight-forward manner from the well defined procedures of the Ruck-Ufimtsev high-frequency scattering technique. Unfortunately, no first-order technique can be used to obtain an accurate formulation of the scattering from surfaces and edges which are partially shadowed or just within the shadow region. The screening functions of the Ruck-Ufimtsev technique have been used in computing the response of several aerospace targets at aspect angles in which some of the target surfaces and edges are shadowed; however, the use of these functions is a first-order approximation, and multiple diffraction and reflection must be considered in order to accurately describe scattered returns from shadowed target regions.

The formulation of the multiple scattering between the surfaces adjacent to the concave edge structures is expressed in the following equations, where the "*" is used to distinguish these expressions from the first order approximations on page L-126:

$$g(1*) = \sqrt{4k}\,A_1^* e^{ip_1^*}$$

$$g(3*) = \sqrt{4k}\,A_3^* e^{ip_3^*}$$

$$g(1\overset{*}{3}) = \sqrt{4k}\,A_{13}^* e^{ip_{13}^*}$$

where $A_1^* = \sqrt{a_1}\,h_1\sin\theta$ for $\theta < \alpha_1$

$= \sqrt{a_1}(a_2-a_1)\cos\theta$ $\theta \geq \alpha_1$

$A_3^* = \sqrt{a_2}\,2h_2\sin\theta$ for $\theta < \alpha_2$

$= \sqrt{a_2}(a_4-a_2)\cos\theta$ $\theta \geq \alpha_2$

$A_{13}{}^* = \sqrt{a_1}\left\{h_1\sin\theta-(a_2-a_1)\cos\theta\right\}$ $\alpha_1 \leq \theta \leq \alpha_3$

$= \sqrt{a_1}\left\{(a_4-a_2)\cos\theta-2h_2\sin\theta\right\}$ $\alpha_3 < \theta < \alpha_2$

and $$p_1^* = -2k(a_1 \sin\theta + h_2 \cos\theta)$$

$$p_3^* = -2k(a_2 \sin\theta - h_2 \cos\theta)$$

$$p_{13}^* = -2k(a_1 \sin\theta - h_2 \cos\theta)$$

This approximate representation of the multiple reflected fields cannot be considered to be an exact or rigorous formulation of the scattering mechanisms involved. Nontheless, this preliminary analytical expression is simple, and the computed results do provide a realistic measure of the target scattering.

L.7.1 Inputs

The subroutine inputs are read from cards or passed in common blocks. The parameters passed in unlabeled common include:

NMIN = minimum frequency sample number

NMAX = maximum frequency sample number

DF = frequency increment (in MHz)

FC = carrier frequency or center frequency (in GHz)

The parameters passed in a labeled common block include:

ASPECT = azimuth angle (in degrees)

ITT = Read data option set to 1 or 2

= 1 Read target dimensions from input card

= 2 Use dimensions input on prior read

The card inputs are the following:

	MATH SYMBOL	VARIABLE	DEFINITION	COLUMNS
Card 1	a_1	A1	Smallest cylinder radius (inches)	1 - 8
	a_2	A2	Middle cylinder radius (inches)	9 - 16
	a_4	A4	Largest cylinder radius (inches)	17 - 24
	h_1	H1	Length of smallest cylinder (inches)	25 - 32
	h_2	H2	Half-length of middle cylinder (inches)	33 - 40
	h_3	H3	Length of largest cylinder (inches)	41 - 48

L.7.2 Outputs

The data base output consists of four linear arrays, EVVR, EVVI, EHHR, EHHI, which contain the real and imaginary parts of the vertically and horizontally polarized backscattered fields (in meters) at frequency increments of DF MHz from NMIN*DF to NMAX*DF.

L.7.3 Restrictions

L.7.3.1 Physical Dimensions

All target dimensions should be large with respect to the wavelength of the illuminating field. In formulating the Q functions to approximate the effects of shadowing on an edge or surface, the assumption that $\alpha_1<\alpha_3<\alpha_2$ was utilized, since this angle relationship was exhibited in the geometry of the target for which the formulation was developed. If this angular relationship is not present, the arguments of the Q functions must be modified.

L.7.3.2 Output

The output arrays are passed in the argument list and a value is computed and stored only in locations NMIN to NMAX.

L.7.3.3 Restrictions

The azimuth angle is restricted to the region between 0 and 180 degrees. In addition, the specular azimuth of 0, 90, and 180 should not be used.

In order to compute the response of these angles, an angular offset of approximately 0.01 degrees should be used.

L.7.4 Definitions of Selected Terms Used in Subroutine

$$SV12 = C(1.5) \mp B(1.5, \pi/2-\theta) \qquad \text{for } \theta < \pi/2$$

$$PH1P = JJ_{11+}$$

$$= JJ_{1m\mp} = J_o(2ka_m \sin\theta) \mp J_1(2ka_m \sin\theta)$$

where $m = 1$ and the lower (+) sign is used

$$BCIX = C(1.5)JJ_{21}$$

where $BC1 = JJ_{21}$

$$= JJ_{2m} = J_o(2ka_m \sin\theta) + J_2(2ka_m \sin\theta)$$

where $m = 1$

$$C11 = e^{ip_{11}}$$

where $p_{11} = - 2k(h_1 + h_2) \cos\theta$

$$CV12 = a_1 e^{ip_{11}} \left\{ JJ_{11+} [C(1.5) \mp B(1.5, \pi/2-\theta)] - C(1.5)JJ_{21} \right\}$$

for $\theta < \pi/2$

$$CV6 = g(6) = a_2 e^{ip_2} \left\{ JJ_{12+} [C(1.5) \mp B(1.5, \pi/2-\theta)] - C(1.5)JJ_{22} \right\} Q_6 IQ_6$$

PSD1 $= p_1^* = -2k(a_1\sin\theta) + h_2\cos\theta)$

AD1 $= A_1^* = \sqrt{a_1}h_1\sin\theta$ for $\theta < \alpha_1$, and

AD1 $= A_1^* = \sqrt{a_1}(a_2-a_1)\cos\theta$ for $\theta \geq \alpha_1$

L.7.5 Subroutines Used

Subfunctions:

1. Q(X) returns the value of the exponential smoothing function.

Subroutines:

1. BESL(XCx, BJx0 BJx1, BJx2) computes the Bessel functions of order 1, 2, and 3 for real argument XCX and returns

   J_o (XCx) in BJx0

   J_1 (XCx) in BJx1

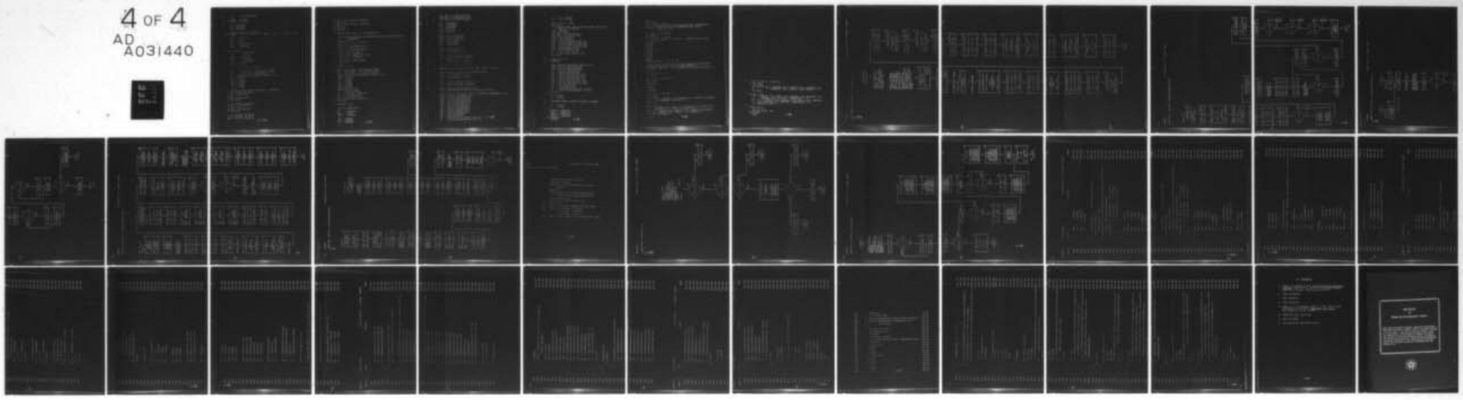
   J_2 (XCx) in BJx2

```
$       IDENT    BECAGD01,HANCOCK,017073100380
$       OPTION   FORTRAN
$       FORTY    LSTIN,XREF,MAP,DECK
$       LIMITS   05,39K,0,5K
      SUBROUTINE TARGET (EVVR, EVVI, EHHR, EHHI)
C
C     **  RESPONSE OF STEPPED CYLINDERS (MODEL 43)     **
C     **  COMPUTED USING THE RUCK UFIMTSEV TECHNIQUE   **
C
      COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, T0
C     NMIN = MINIMUM FREQUENCY SAMPLE
C     NMAX = MAXIMUM FREQUENCY SAMPLE
C     DF   = FREQUENCY INCREMENT IN MHZ
C     FC   = CARRIER FREQUENCY IN GHZ
      COMMON /TARS / ASPECT,ITT
C     ITT SET TO 1 TO READ INPUT DIMENSIONS
C         SET TO 2 WHEN TARGET DIMENSIONS DO NOT CHANGE
C
      COMPLEX PH1, PH2, PH4, PH1P, PH2P, PH4P, C11, C2, C4, C9,
     1        PH5P, PH7P
      COMPLEX  SV,SH
      COMPLEX CV12,CH12, CV11,CH11, CV2,CH2, CV4,CH4, CV9,CH9,
     1        CV10,CH10, CV1, CV3, CV6,CH6, CV8,CH8, CV5, CV7
      COMPLEX CD1, CD2, CD3, CAD
      DIMENSION EVVR(1), EVVI(1), EHHR(1), EHHI(1)
C
      GO TO ( 5,6 ), ITT
    5 CONTINUE
C     PROGRAM CONSTANTS
C
      PI          = 3.14159265358979
      FTK         =  53234454
      FTKDF    =  53234454*(DF/1000.)
      PIO2        = PI/2.
      SPI         = SQRT(PI)
      AITM        = 0.0254026
      SPIK = SPI*AITM
      TPIO2       = 3.*PIO2
      DTR         = PI/180.
      RTD         = 180./PI
      DLT      = 15.*DTR
C
      READ (5,5001)           A1, A2, A4, H1, H2, H3
 5001 FORMAT ( 7F8.0)
      WRITE (6,6001) ASPECT
 6001 FORMAT ( '1  PROGRAM INPUT PARAMETERS',//,' THETA =',F9.3 )
      WRITE (6,6002) A1,H1,A2,H2,A4,H3
 6002 FORMAT ( '0  A1 =', F7.4,'   H1 =',F7.4,/,
     1          '   A2 =', F7.4,'   H2 =',F7.4,/,
     2          '   A4 =', F7.4,'   H3 =',F7.4 )
C
      TH2         = H2+H2
      TH2PH1      = TH2+H1
      A21         = A2-A1
      A41         = A4-A1
      A42         = A4-A2
      SA1   = SQRT(A1)
      SA2   = SQRT(A2)
C
      ALF1      = ATAN(A21/H1)
```

AD-A031 440   GENERAL DYNAMICS FORT WORTH TEX CONVAIR AEROSPACE DIV   F/G 17/9
ENDO ATMOSPHERIC EXO ATMOSPHERIC RADAR MODELING. RADAR CROSS SE--ETC(U)
JUN 76 R J HANCOCK, F H CLEVELAND   F30602-73-C-0380

UNCLASSIFIED   RADC-TR-76-186-VOL-4-PT-2   NL

4 OF 4
AD
A031440



END

DATE
FILMED
12-76

```
      ALF3     = ATAN(A41/TH2PH1)
C
      PIMAL2   = PI-ALF2
      PIMAL3   = PI-ALF3
C
      X1D = ALF1*RTD
      X2D = ALF2*RTD
      X3D = ALF3*RTD
C
      WRITE (6,6005) X1D,X2D,X3D
 6005 FORMAT ( '   ALF1 = ',F8.3,/,'   ALF2 = ',F8.3,/,'   ALF3 = ',F8.3)
C
      TKFC     = FTK*FC*2.
      XN       = 1.5
      XNO2     = XN/2.
      SOXN     = SIN(PI/XN)/XN
      CXN      = -SOXN/XN
      CPON     = COS(PI/XN)
C
    6 CONTINUE
      TH            = ASPECT*DTR
      STH           = SIN(TH)
      CTH           = COS(TH)
      TANTH         = STH/CTH
      CK1           = 0.5*TANTH
      CKV           = 0.5/TANTH
      C1PCV = CK1+CKV
C
      IF ( TH .GT. PIO2) GO TO 20
C
      B12      = SOXN/(CPON - COS((PIO2-TH)/XNO2))
      B11      = SOXN/(CPON - COS((PIO2+TH)/XNO2))
      B9       = SOXN/(CPON - COS(TH/XNO2))
C     EDGE DIFFRACTION COEFFICIENTS (THETA.LT.PI/2)
      SV12 = CXN-B12
      SH12 = CXN+B12
      SV11 = CXN-B11
      SH11 = CXN+B11
      SV9  = CXN-B9
      SH9  = CXN+B9
C
C     EFFECTIVE AREA (DOUBLY REFLECTED ) TERMS(ADX)
      IF (TH-ALF1) 51,52,52
   51 AD1  = SA1*H1*STH
      GO TO 53
   52 AD1  = SA1*A21*CTH
   53 CONTINUE
C
      IF (TH-ALF2) 54,55,55
   54 AD2  = SA2*TH2*STH
      GO TO 56
   55 AD2  = SA2*A42*CTH
   56 CONTINUE
C
      IF (TH-ALF1) 57,58,58
   58 IF (TH-ALF2) 59,59,57
   59 IF (TH-ALF3) 60,61,61
```

```
60 AD3   = SA1*(H1*STH - A21*CTH)
   GO TO 63
61 AD3   = SA1*(A42*CTH - TH2*STH)
   GO TO 63
57 AD3 = 0
63 CONTINUE

   ASSUMED THAT  ALF3 .GT. (ALF1 OR ALF2 )

   SMOOTHING FUNCTIONS AND ILLUMINATED AREA USING SLIDING SHADOW
            BOUNDARIES
   ISWA3   = 1
   IF ( TH .GT. ALF3 ) GO TO 25
   ISWA3   = 2
     Q6    = Q(TKFC*A2*(ALF1-TH))
   IF (Q6 .LE. 0.5 ) Q6 = 0.
     Q8    = Q(TKFC*A4*(ALF3-TH))
   IF (Q8 .LE. 0.5 ) Q8 = 0.
     AS5   = A1 + H1*TANTH
     IF ( TH .GT. ALF1 ) GO TO 14
     AS7   = A2 + TH2*TANTH
     GO TO 25
14   AS7   = A1 + TH2PH1*TANTH
     GO TO 25

   THETA .GT. PI/2

20 B10 = SOXN/(CPON - COS((TH-PIO2)/XNO2))
   B9  = SOXN/(CPON - COS((TPIO2-TH)/XNO2))
   B11 = SOXN/(CPON - COS((PI-TH)/XNO2))
   EDGE DIFFRACTION COEFFICIENTS (THETA.GT.PI/2)
   SV10 = CXN-B10
   SH10 = CXN+B10
   SV9  = CXN-B9
   SH9  = CXN+B9
   SV11 = CXN-B11
   SH11 = CXN+B11
   XALF = PIO2+DLT-TH
   SMOOTHING FUNCTIONS
   Q1    = Q(TKFC*A1*XALF)
   Q2   = Q(TKFC*A2*(PIMAL2-TH))
   Q3    = Q(TKFC*A2*XALF)
   Q4   = Q(TKFC*A4*(PI-TH))
   Q11 = Q(TKFC*A1*(PIMAL3-TH))
25 CONTINUE

   FREQUENCY LOOP

   DO 30 I = NMIN,NMAX
   XI      = I-1
   AK      = XI*FTKDF
   AK2     = AK+AK
   AK2S    = AK2*STH

   XC1   = A1*AK2S
   XC2   = A2*AK2S
   XC4   = A4*AK2S
```

```
      CALL BESL(XC1,BJ10,BJ11,BJ12)
      CALL BESL(XC2,BJ20,BJ21,BJ22)
      CALL BESL(XC4,BJ40,BJ41,BJ42)
C
      BC1  = BJ10+BJ12
      BC2  = BJ20+BJ22
      BC4  = BJ40+BJ42
      BC1X = BC1*CXN
      BC2X = BC2*CXN
      BC4X = BC4*CXN
C
      PH1P = CMPLX(BJ10,BJ11)
      PH2P = CMPLX(BJ20,BJ21)
      PH4P = CMPLX(BJ40,BJ41)
      PH1  = CONJG(PH1P)
      PH2  = CONJG(PH2P)
      PH4  = CONJG(PH4P)
C
      PS11 = -AK2*(H2+H1)*CTH
      PS2  = -AK2*H2*CTH
      PS9  =  AK2*(H2+H3)*CTH
C
      C11  = CMPLX(COS(PS11),SIN(PS11))
      C2   = CMPLX(COS(PS2),SIN(PS2))
      C4   = CONJG(C2)
      C9   = CMPLX(COS(PS9),SIN(PS9))
C
      IF ( TH .GT. PIO2 ) GO TO 35
C
C     GENERAL REGION (THETA .GT. ALF3 .AND. THETA .LT. PI/2)
C
C     PHASE OF DOUBLY REFLECTED  SCATTERING TERMS
      PSD1 = -AK2*(A1*STH + H2*CTH)
      PSD2 = -AK2*(A2*STH - H2*CTH)
      PSD3 = -AK2*(A1*STH - H2*CTA)
C
      CD1  = CMPLX(COS(PSD1),SIN(PSD1))
      CD2  = CMPLX(COS(PSD2),SIN(PSD2))
      CD3  = CMPLX(COS(PSD3),SIN(PSD3))
C
C     DOUBLY REFLECTED SCATTERING (TOTAL)
      CAD  = SQRT(AK2+AK2)*(AD1*CD1 + AD2*CD2 + AD3*CD3)
C
C     RUCK-UFIMTSEV SCATTERING (THETA.LT.90,EDGES ILLUMINATED AT 90)
      CV12 = A1*C11*(SV12*PH1P-BC1X)
      CH12 = A1*C11*(SH12*PH1P-BC1X)
      CV11 = A1*C11*(SV11*PH1-BC1X)
      CH11 = A1*C11*(SH11*PH1-BC1X)
      CV2  = A2*C2*(SV11*PH2-BC2X)
      CH2  = A2*C2*(SH11*PH2-BC2X)
      CV4  = A4*C4*(SV11*PH4-BC4X)
      CH4  = A4*C4*(SH11*PH4-BC4X)
      CV9  = A4*C9*(SV9*PH4-BC4X)
      CH9  = A4*C9*(SH9*PH4-BC4X)
      CV1  = A1*C2*PH1*C1PCV
      CV3  = A2*C4*PH2*C1PCV
      SV   = CV12+CV11+CV2+CV4+CV9 +CV1 +CV3
      SH   = CH12+CH11+CH2+CH4+CH9 -CV1 -CV3
```

```
C
      SV    = SV + CAD/SPI
      SH    = SH + CAD/SPI
C
      GO TO (40,34),  ISWA3
   34 CONTINUE
C     AZIMUTH NEAR ZERO, ADD IN FAR EDGE TERMS WHICH ARE
C           DIRECTLY ILLUMINATED
      XCS5 = AS5*AK2S
      XCS7 = AS7*AK2S
      CALL BESL(XCS5,B50,B51,B52)
      CALL BESL(XCS7,B70,B71,B72)
      PH5P = CMPLX(B50,B51)
      PH7P = CMPLX(B70,B71)
      CV6   = A2*C2*(PH2P*SV12-BC2X)*Q6
      CH6   = A2*C2*(PH2P*SH12-BC2X)*Q6
      CV8   = A4*C4*(PH4P*SV12-BC4X)*Q8
      CH8   = A4*C4*(PH4P*SH12-BC4X)*Q8
      CV5   =-AS5*C2*PH5P*CKV*Q6
      CV7   =-AS7*C4*PH7P*CKV*Q8
      SV    = CV6 + CV8 +CV5+CV7 + SV
      SH    = CH6 + CH8 -CV5-CV7 + SH
      GO TO 40
C
C     THETA .GT. PI/2
   35 CONTINUE
C
C     RUCK-UFIMTSEV SCATTERING (THETA.GT.90)
      CV10 = A4*C9*(PH4P*SV10-BC4X)
      CH10 = A4*C9*(PH4P*SH10-BC4X)
      CV9   = A4*C9*(PH4*SV9-BC4X)
      CH9   = A4*C9*(PH4*SH9-BC4X)
      CV4   = A4*C4*(PH4*SV11-BC4X)*Q4
      CH4   = A4*C4*(PH4*SH11-BC4X)*Q4
      CV2   = A2*C2*(PH2*SV11-BC2X)*Q2
      CH2   = A2*C2*(PH2*SH11-BC2X)*Q2
      CV11 = A1*C11*(PH1*SV11-BC1X)*Q11
      CH11 = A1*C11*(PH1*SH11-BC1X)*Q11
      CV1   = A1*C2*PH1*CK1*Q1
      CV3   = A2*C4*PH2*CK1*Q3
      SV    = CV10+CV9+CV4+CV2+CV11 +CV1+CV3
      SH    = CH10+CH9+CH4+CH2+CH11 -CV1-CV3
   40 CONTINUE
C
      SV =-SV*SPIK
      SH = SH*SPIK
C
C     REFERENCE PHASE TO FRONT OF SECOND CYLINDER
C
      SV    = SV*C4
      SH    = SH*C4
C
      EVVR(I) = REAL(SV)
      EVVI(I) =-AIMAG(SV)
      EHHR(I) = REAL(SH)
      EHHI(I) =-AIMAG(SH)
   30 CONTINUE
      RETURN
      END
      FUNCTION Q(Z)
C     Q(Z) = 0.5*(1 + ERF(Z))
C   * ERF(Z) IS EVALUATED USING A RATIONAL POLYNOMIAL APPROXIMATION
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,
C   *               SECTION 7.1.26)
C
      IF ( Z.GT. 2. ) GO TO 10
      IF ( Z.LT.-2. ) GO TO 20
      AZ = ABS(Z)
      P = 1.0/(1.0 + .47047*AZ)
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)
      IF (Z) 2,4,6
    2 Q = (1.0 - Y)/2.
      RETURN
    4 Q = .5
      RETURN
    6 Q = (1.0 + Y)/2.
      RETURN
   10 Q = 1.
      RETURN
   20 Q = 0.
      RETURN
      END
      SUBROUTINE BESL ( X, B0, B1, B2 )
C
C   * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS
C   * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS
C   * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )
C
      S = 1.0
      IF (X .LT. 0.0) S=-1.0
      X = ABS(X)
C
      IF ( X .GT. 1.E-6 ) GO TO 5
      B0 = 1.0
      B1 = 0.0
      B2 = 0.0
      X = X * S
      RETURN
C
    5 CONTINUE
C
    1 IF ( X .GE. 3. ) GO TO 9
      X1 = X/3.
      X1 = X1*X1
      B = 1. + X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479
     1 +       X1*(-.0039444+ X1*2.1E-4 )))) )
      GO TO 10
C
    9 X2 = 3./X
      F0 =   .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*
     1      (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )
      T0 = X - .78539816 +X2*(-.04166397 +X2*(-.3954E-4 +X2*(.00262573
     1    +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))
      B = F0*COS(T0)/SQRT(X)
C
   10 B0 = B
```

```
C
   2 IF ( X . GE.  3.  ) GO TO 19
     X1 = X/3.
     X1 = X1*X1
     B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*
    1                (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )
     GO TO 20
C
  19 X2 = 3./X
     F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*
    1       (-.00249511 +X2*(.00113653 -.00020033*X2 )))))
     T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879
    1     +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))
     B = F1*COS(T1)/SQRT(X)
C
  20 B1 = B * S
     X = X * S
     B2= (2./X)*B1 - B0
  50 RETURN
     END
```

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,FFF [illegible])

L-142

```
     -----------
    / TARGET  /
     -----------

          *               +------------------->*
                          |                    |  NOTE  08
 ** RESPONSE OF           |         * * * * * * * * * * *
STEPPED CYLINDERS         |         *  LIST = A1, A2,    *
(MODEL 43)     **         |         *  A4, H1, H2, H3    *
 ** COMPUTED USING        |         * * * * * * * * * * *
THE RUCK UFIMTSEV         |                    |
TECHNIQUE  **             |                    |
                          |                    |       09
                          |              ---------------
                          |            / WRITE TO DEV  /
NMIN = MINIMUM            |           /       6       /
FREQUENCY SAMPLE          |          /  VIA FORMAT   /
NMAX = MAXIMUM            |         /     6001      /
FREQUENCY SAMPLE          |        / FROM THE LIST /
DF   = FREQUENCY          |        ---------------
INCREMENT IN MHZ          |                    |
FC   = CARRIER            |                    |
FREQUENCY IN GHZ          |                    |  NOTE  10
ITT SET TO 1 TO READ      |         * * * * * * * * * * *
INPUT DIMENSIONS          |         *   LIST = ASPECT   *
SET TO 2 WHEN TARGET      |         * * * * * * * * * * *
DIMENSIONS DO NOT         |                    |
CHANGE                    |                    |
                          |                    |       11
          *               |              ---------------
          |         02    |            / WRITE TO DEV  /
*-------------------*     |           /       6       /
|  COMPUTED CO TO   |     |          /  VIA FORMAT   /
|     FOR  ITT      |     |         /     6002      /
*-------------------*     |        / FROM THE LIST /
| 5          50.03  |     |        ---------------
| 6          51.05  |     |                    |
*-------------------*     |                    |
          |               |                    |  NOTE  12
          |               |         * * * * * * * * * * *
          |               |         *  LIST = A1, H1,    *
IF OUTSIDE THE RANGE      |         *  A2, H2, A4, H3    *
```

```
| 6            51.05 |
*--------------------*
          |
          |
          |
IF OUTSIDE THE RANGE
          |
 50.02--->|
    5     |  NOTE  03
* * * * * * * * * * * *
*      CONTINUE       *
* * * * * * * * * * * *
          |
          |
          |
PROGRAM CONSTANTS
          |
          |
          |            04
*--------------------*
|       PI =         |
| 3.14159265358979   |
|                    |
|  FTK = .53234454   |
|                    |
|      FTKDF =       |
|  .53234454*(DF/    |
|      1000.)        |
|                    |
|   PIO2 = PI/2.     |
*--------------------*
          |
          |
          |            05
*--------------------*
|  SPI = SQRT(PI)    |
|                    |
| AITM = 0.0254026   |
|                    |
|  SPIK = SPI*AITM   |
|                    |
|  TPIO2 = 3.*PIO2   |
*--------------------*
          |
          |
          |            06
*--------------------*
|   DTR = PI/180.    |
|                    |
|   RTD = 180./PI    |
|                    |
|   DLT = 15.*DTR    |
*--------------------*
          |
```

```
          |
          |
          |  NOTE  12
* * * * * * * * * * * *
*  LIST = A1, H1,     *
*  A2, H2, A4, H3     *
* * * * * * * * * * * *
          |
          |
          |            13
*--------------------*
|   TH2 = H2 + H2    |
|                    |
| TH2PH1 = TH2 + H1  |
|                    |
|   A21 = A2 - A1    |
|                    |
|   A41 = A4 - A1    |
*--------------------*
          |
          |
          |            14
*--------------------*
|   A42 = A4 - A2    |
|                    |
|  SA1 = SQRT(A1)    |
|                    |
|  SA2 = SQRT(A2)    |
*--------------------*
          |
          |
          |            15
*--------------------*
|      ALF1 =        |
|   ATAN(A21/H1)     |
|                    |
|      ALF2 =        |
|   ATAN(A42/TH2)    |
|                    |
|      ALF3 =        |
| ATAN(A41/TH2PH1)   |
*--------------------*
          |
          |
          |            16
*--------------------*
|  PIMAL2 = PI -     |
|      ALF2          |
|                    |
|  PIMAL3 = PI -     |
|      ALF3          |
*--------------------*
          |
```

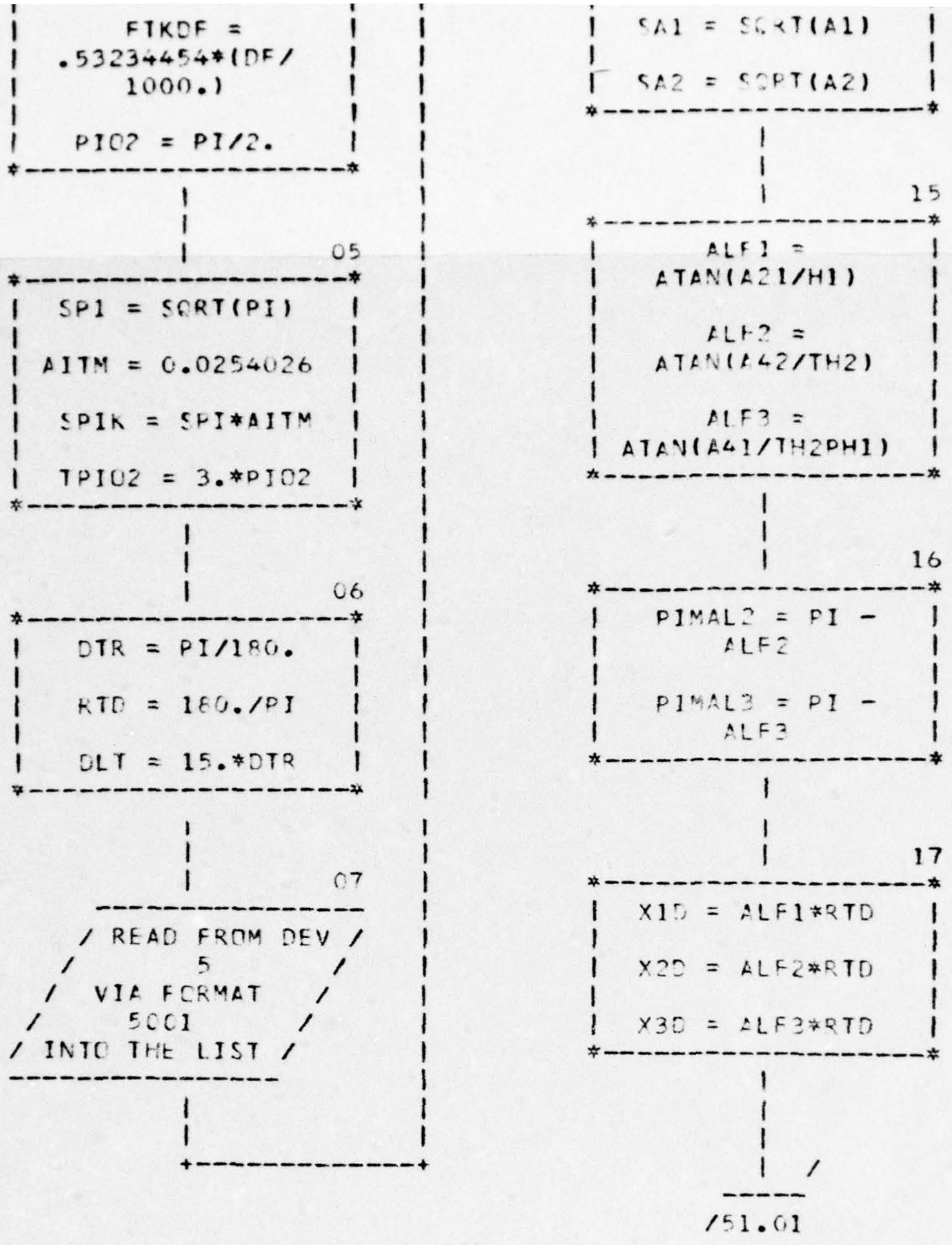
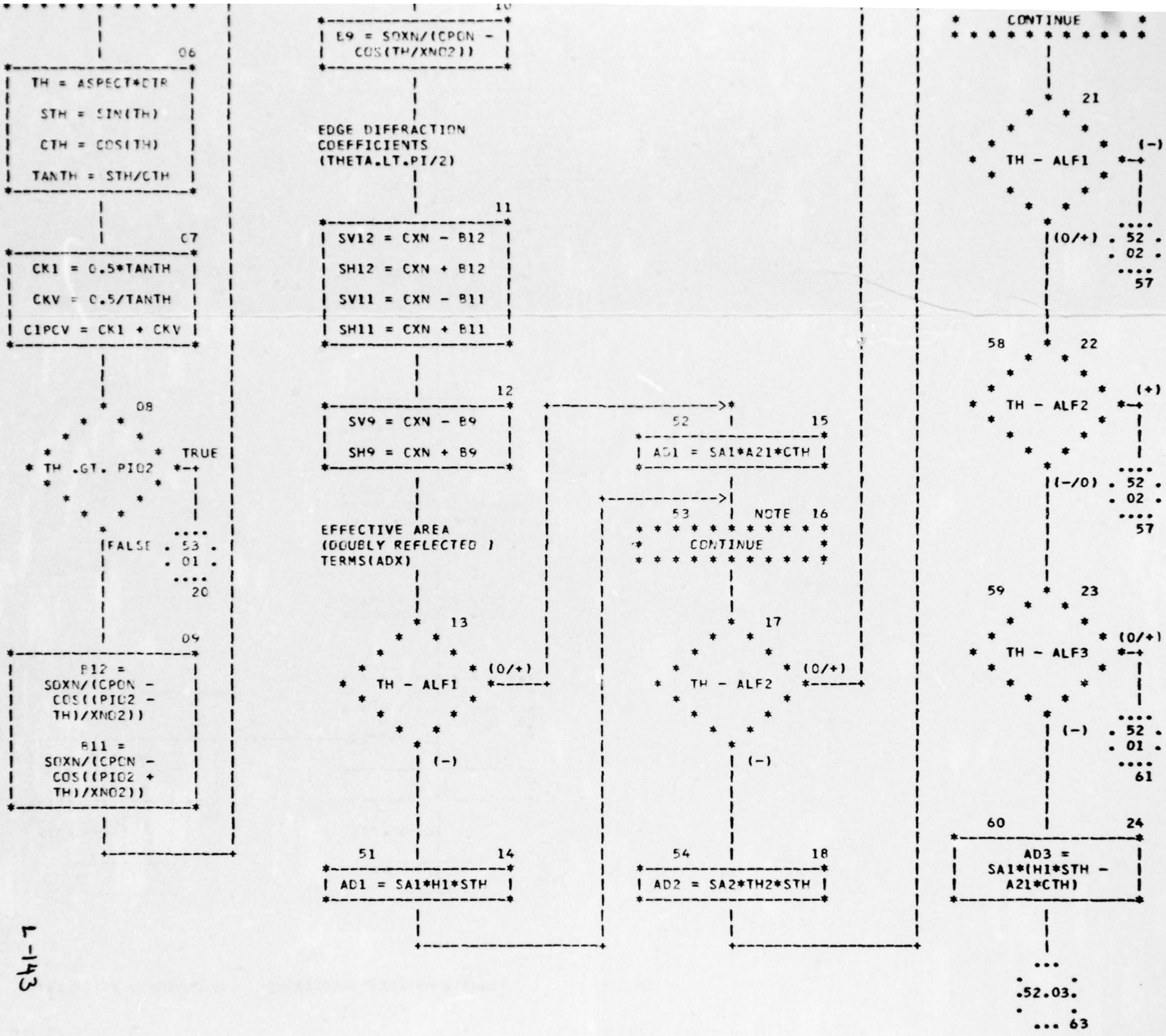

FTKDF =
.53234454*(DF/
1000.)
PIO2 = PI/2.
05
SP1 = SQRT(PI)
AITM = 0.0254026
SPIK = SPI*AITM
TPIO2 = 3.*PIO2
06
DTR = PI/180.
RTD = 180./PI
DLT = 15.*DTR
07
READ FROM DEV
5
VIA FORMAT
5001
INTO THE LIST
SA1 = SQRT(A1)
SA2 = SQRT(A2)
15
ALF1 =
ATAN(A21/H1)
ALF2 =
ATAN(A42/TH2)
ALF3 =
ATAN(A41/TH2PH1)
16
PIMAL2 = PI -
ALF2
PIMAL3 = PI -
ALF3
17
X1D = ALF1*RTD
X2D = ALF2*RTD
X3D = ALF3*RTD
/51.01

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI)

```
    50.17--->*
             |          01
       -------------------
       / WRITE TO DEV  /
      /       6       /
     /  VIA FORMAT   /
    /      6005     /
   / FROM THE LIST /
   -------------------
             |
             |
             |  NOTE    02
  * * * * * * * * * * * * *
  * LIST = X1D, X2D,      *
  *        X3D            *
  * * * * * * * * * * * * *
             |
             |
             |          03
  *------------------------*
  | TKFC = FTK*FC*2.       |
  |                        |
  |     XN = 1.5           |
  |                        |
  |   XN02 = XN/2.         |
  |                        |
  |      SOXN =            |
  |    SIN(PI/XN)/XN       |
  |                        |
  |   CXN = - SOXN/XN      |
  *------------------------*
             |
             |
             |          04                                                +-------------------->*
  *------------------------*                                              |               55     |         19
  | CPON = COS(PI/XN)      |                                              |         *-----------------------*
  *------------------------*                                              |         | AD2 = SA2*A42*CTH     |
             |                                                            |         *-----------------------*
    50.02--->|                                                            |                     |
     6       |  NOTE    05                                                |     +-------------->|
  * * * * * * * * * * * * *                                               |     |         56    |  NOTE   20
  *        CONTINUE       *   +--------------------->*                    |     |   * * * * * * * * * * * * *
  * * * * * * * * * * * * *   |                      |          10        |     |   *        CONTINUE       *
             |                |          *-----------------------*        |     |   * * * * * * * * * * * * *
             |                |          | E9 = SOXN/(CPON -     |        |     |               |
             |          06    |          |   COS(TH/XN02))       |        |     |               |
  *------------------------*  |          *-----------------------*        |     |               |
  | TH = ASPECT*DTR        |  |                      |                    |     |               |
```

2


06
TH = ASPECT*DTR
STH = SIN(TH)
CTH = COS(TH)
TANTH = STH/CTH
C7
CK1 = 0.5*TANTH
CKV = 0.5/TANTH
C1PCV = CK1 + CKV
08
TH .GT. PIO2
TRUE
FALSE
53
01
20
09
B12 =
SOXN/(CPON -
COS((PIO2 -
TH)/XNO2))
B11 =
SOXN/(CPON -
COS((PIO2 +
TH)/XNO2))
10
E9 = SOXN/(CPON -
COS(TH/XNO2))
EDGE DIFFRACTION
COEFFICIENTS
(THETA.LT.PI/2)
11
SV12 = CXN - B12
SH12 = CXN + B12
SV11 = CXN - B11
SH11 = CXN + B11
12
SV9 = CXN - B9
SH9 = CXN + B9
EFFECTIVE AREA
(DOUBLY REFLECTED)
TERMS(ADX)
13
TH - ALF1
(0/+)
(-)
51
14
AD1 = SA1*H1*STH
52
15
AD1 = SA1*A21*CTH
53
NOTE
16
CONTINUE
17
TH - ALF2
(0/+)
(-)
54
18
AD2 = SA2*TH2*STH
CONTINUE
21
TH - ALF1
(-)
(0/+)
52
02
57
58
22
TH - ALF2
(+)
(-/0)
52
02
57
59
23
TH - ALF3
(0/+)
(-)
52
01
61
60
24
AD3 =
SA1*(H1*STH -
A21*CTH)
52.03.
63

L-143

```
04/26/76                                                      AUTOFLOW CHART SET - FWD/SCL  RADSIM

CHART TITLE - SUBROUTINE  TARGET(EVVR,EVVI,EHHR,EHHI)
```

L-144

```
     -----------                     -----------
    /    61    /                    /    57    /
     -----------                     -----------
          |                               |
 51.23--->|                     51.21*--->|
          |              01               |             02
*--------------------*            *-------------------*
|       AD3 =        |            |      AD3 = 0      |
| SA1*(A42*CTH -     |            *-------------------*
|     TH2*STH)       |                    |
*--------------------*   +-----------51.24*-->|
          |              |           63   |  NOTE  03
          |              |          * * * * * * * * * * * *
          +--------------+          *       CONTINUE      *
                                    * * * * * * * * * * * *
                                                |
                                                |
                                    ASSUMED THAT  ALF3
                                    .GT. (ALF1 OR ALF2 )
                                                |
                                                |
                                    SMOOTHING FUNCTIONS
                                    AND ILLUMINATED AREA
                                    USING SLIDING SHADOW
                                    BOUNDARIES
                                                |
                                                |          04
                                    *-------------------*
                                    |     ISWA3 = 1     |
                                    *-------------------*
                                                |
                                                |
                                                *      05
                                             *     *
                                          *           *
                                        *               *  TRUE
                                      * TH  .GT. ALF3    *--+
                                        *               *   |
                                          *           *     |
                                             *     *        |
                                                *         ....
                                                |FALSE  . 53 .
                                                |       . 07 .
                                                |         ....
                                                |          25
                                                |
                                                |          06
                                    *-------------------*
                                    |     ISWA3 = 2     |     +------------------>*
                                    |                   |
```

2

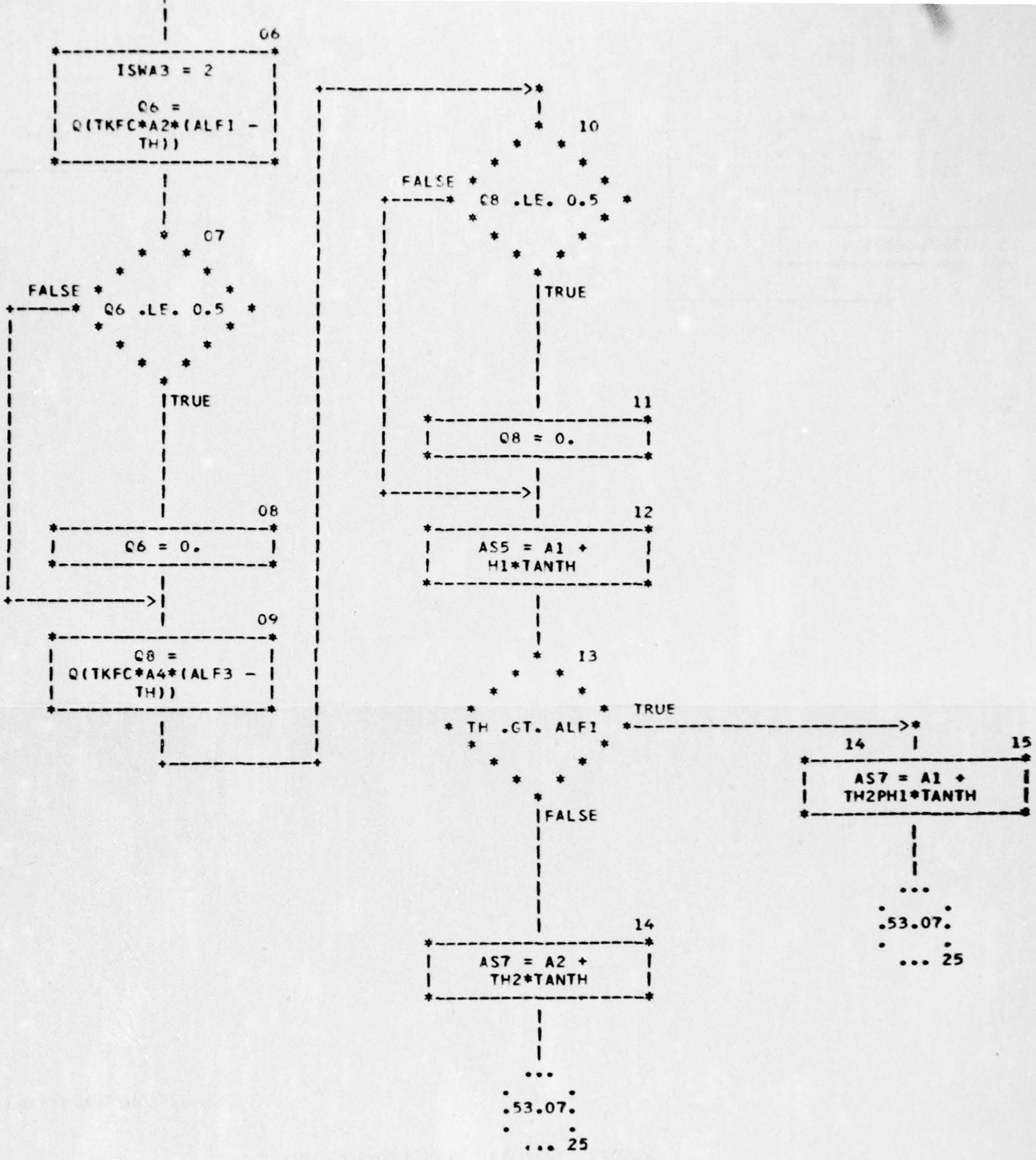
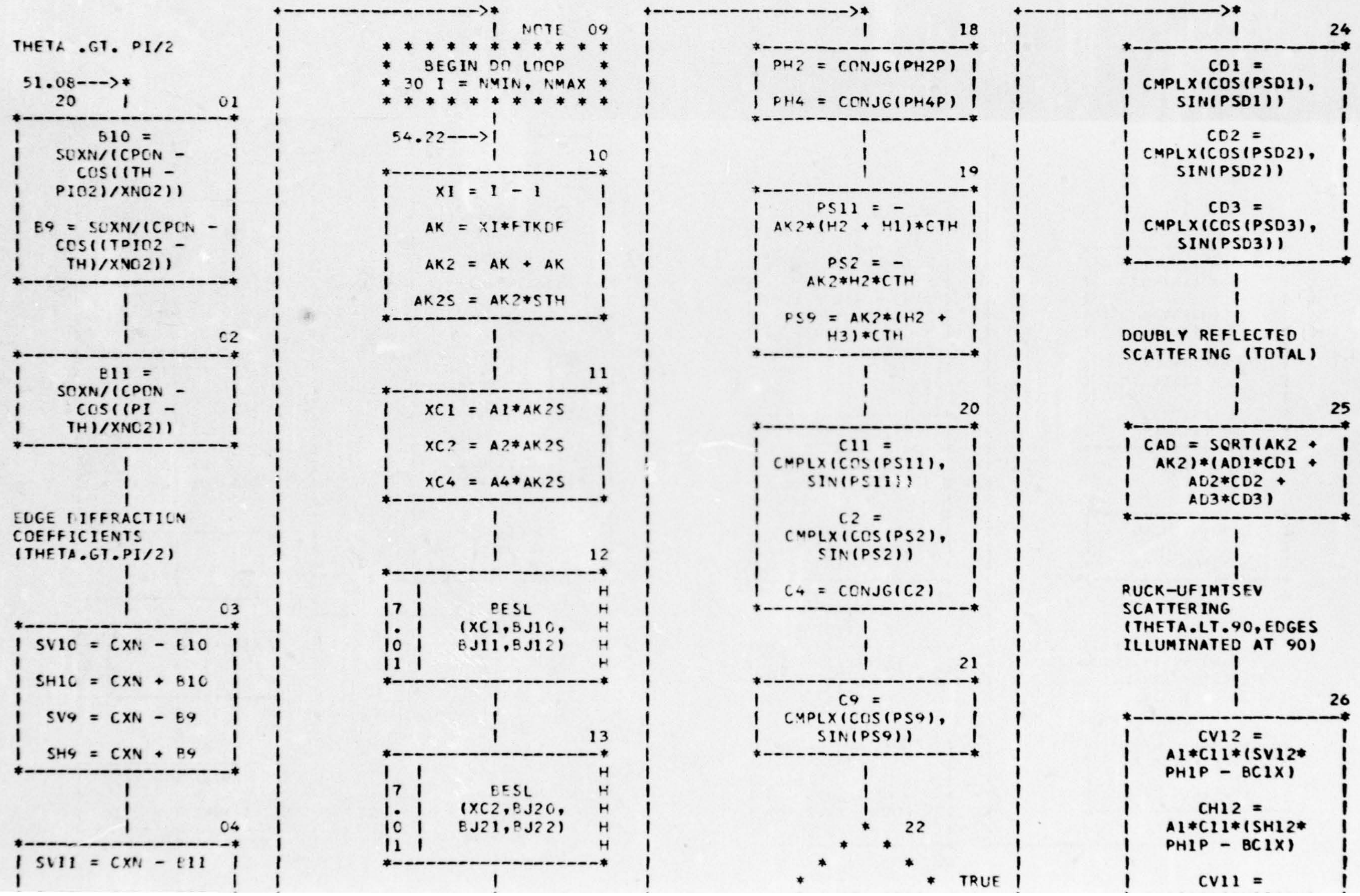
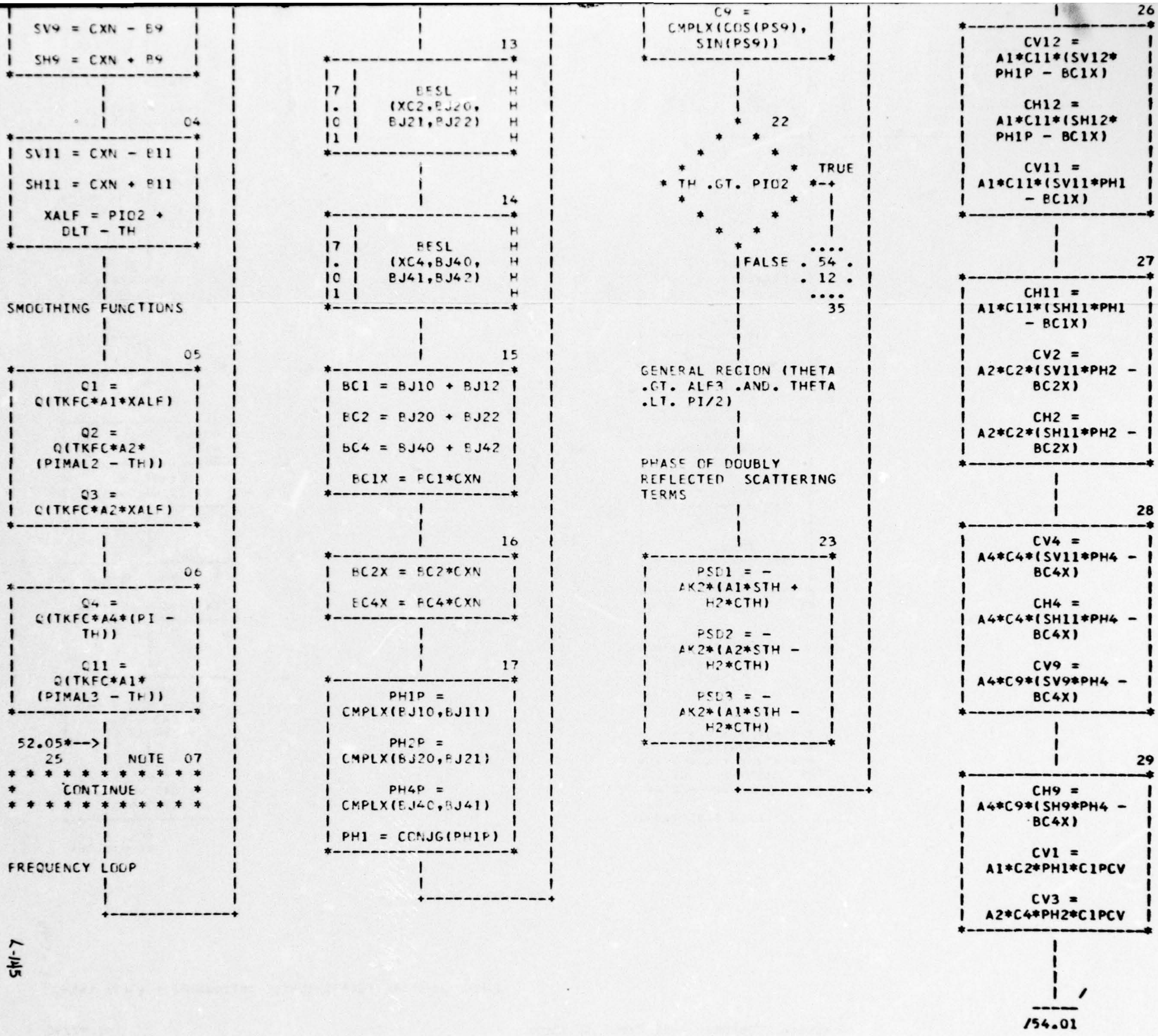

06
ISWA3 = 2
Q6 =
Q(TKFC*A2*(ALFI -
TH))
07
Q6 .LE. 0.5
FALSE
TRUE
08
Q6 = 0.
09
Q8 =
Q(TKFC*A4*(ALF3 -
TH))
10
Q8 .LE. 0.5
FALSE
TRUE
11
Q8 = 0.
12
AS5 = A1 +
H1*TANTH
13
TH .GT. ALFI
TRUE
FALSE
14
AS7 = A2 +
TH2*TANTH
.53.07.
... 25
14
15
AS7 = A1 +
TH2PH1*TANTH
.53.07.
... 25

CHART TITLE - SUBROUTINE TARGET(EVVR,EVVI,EHHR,EHHI)


THETA .GT. PI/2
51.08--->*
20
01
B10 = SOXN/(CPON - COS((TH - PIO2)/XNO2))
B9 = SOXN/(CPON - COS((TPIO2 - TH)/XNO2))
02
B11 = SOXN/(CPON - COS((PI - TH)/XNO2))
EDGE DIFFRACTION COEFFICIENTS (THETA.GT.PI/2)
03
SV10 = CXN - B10
SH10 = CXN + B10
SV9 = CXN - B9
SH9 = CXN + B9
04
SV11 = CXN - B11
NOTE 09
BEGIN DO LOOP
30 I = NMIN, NMAX
54.22--->
10
XI = I - 1
AK = XI*FTKDF
AK2 = AK + AK
AK2S = AK2*STH
11
XC1 = A1*AK2S
XC2 = A2*AK2S
XC4 = A4*AK2S
12
7.01
BESL (XC1,BJ10, BJ11,BJ12)
13
7.01
BESL (XC2,BJ20, BJ21,BJ22)
18
PH2 = CONJG(PH2P)
PH4 = CONJG(PH4P)
19
PS11 = - AK2*(H2 + H1)*CTH
PS2 = - AK2*H2*CTH
PS9 = AK2*(H2 + H3)*CTH
20
C11 = CMPLX(COS(PS11), SIN(PS11))
C2 = CMPLX(COS(PS2), SIN(PS2))
C4 = CONJG(C2)
21
C9 = CMPLX(COS(PS9), SIN(PS9))
22
TRUE
24
CD1 = CMPLX(COS(PSD1), SIN(PSD1))
CD2 = CMPLX(COS(PSD2), SIN(PSD2))
CD3 = CMPLX(COS(PSD3), SIN(PSD3))
DOUBLY REFLECTED SCATTERING (TOTAL)
25
CAD = SQRT(AK2 + AK2)*(AD1*CD1 + AD2*CD2 + AD3*CD3)
RUCK-UFIMTSEV SCATTERING (THETA.LT.90,EDGES ILLUMINATED AT 90)
26
CV12 = A1*C11*(SV12* PH1P - BC1X)
CH12 = A1*C11*(SH12* PH1P - BC1X)
CV11 =



```
SV9 = CXN - B9
SH9 = CXN + B9

04
SV11 = CXN - B11
SH11 = CXN + B11
XALF = PIO2 +
DLT - TH

SMOOTHING FUNCTIONS

05
Q1 =
Q(TKFC*A1*XALF)
Q2 =
Q(TKFC*A2*
(PIMAL2 - TH))
Q3 =
Q(TKFC*A2*XALF)

06
Q4 =
Q(TKFC*A4*(PI -
TH))
Q11 =
Q(TKFC*A1*
(PIMAL3 - TH))

52.05*-->
25   NOTE 07
CONTINUE

FREQUENCY LOOP
```

```
13
7.01  BESL
(XC2,BJ20,
BJ21,BJ22)

14
7.01  BESL
(XC4,BJ40,
BJ41,BJ42)

15
BC1 = BJ10 + BJ12
BC2 = BJ20 + BJ22
BC4 = BJ40 + BJ42
BC1X = BC1*CXN

16
BC2X = BC2*CXN
BC4X = BC4*CXN

17
PH1P =
CMPLX(BJ10,BJ11)
PH2P =
CMPLX(BJ20,BJ21)
PH4P =
CMPLX(BJ40,BJ41)
PH1 = CONJG(PH1P)
```

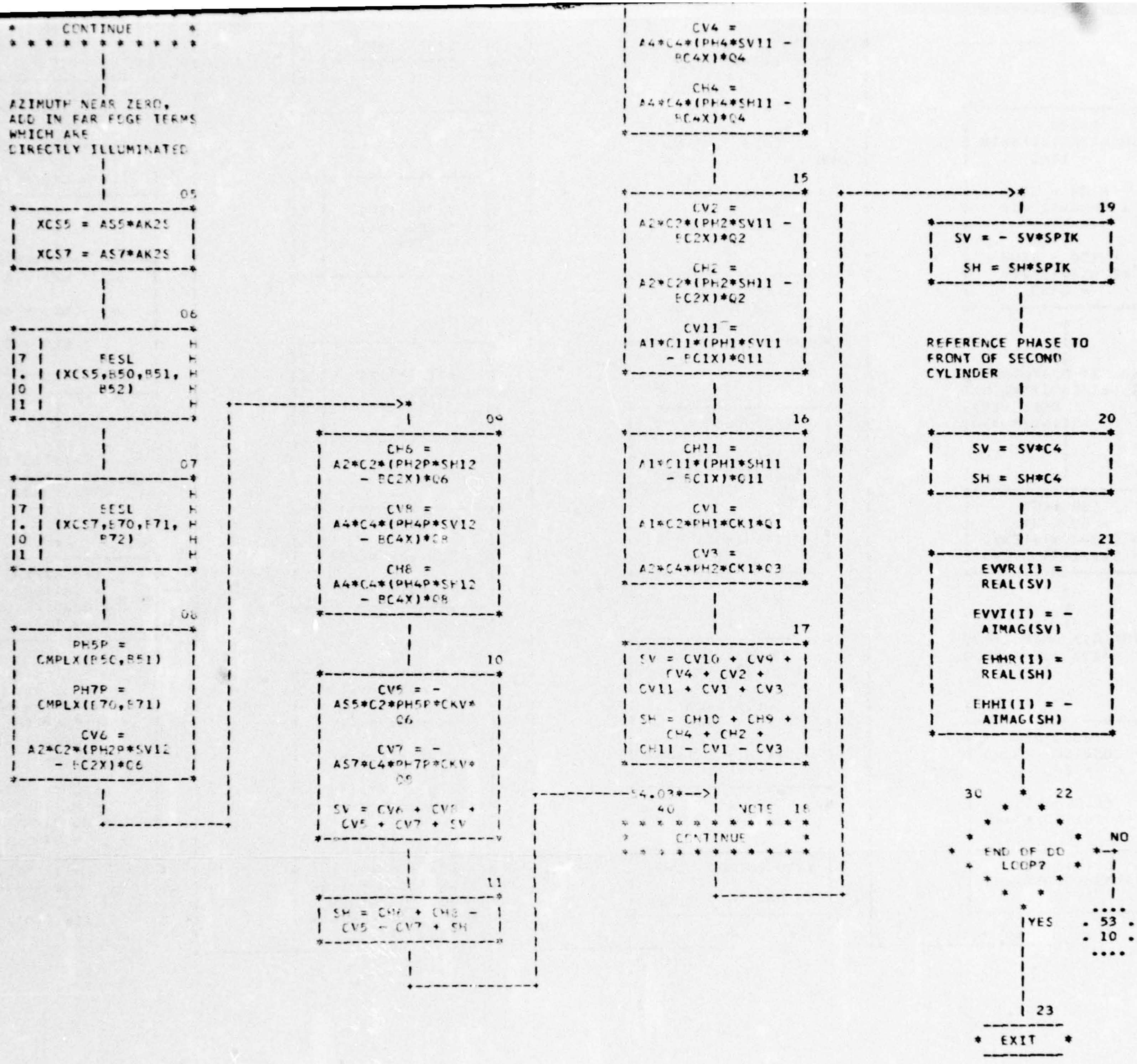
```
C9 =
CMPLX(COS(PS9),
SIN(PS9))

22
TH .GT. PIO2   TRUE
FALSE   54
        12
        35

GENERAL REGION (THETA
.GT. ALF3 .AND. THETA
.LT. PI/2)

PHASE OF DOUBLY
REFLECTED  SCATTERING
TERMS

23
PSD1 = -
AK2*(A1*STH +
H2*CTH)
PSD2 = -
AK2*(A2*STH -
H2*CTH)
PSD3 = -
AK2*(A1*STH -
H2*CTH)
```

```
26
CV12 =
A1*C11*(SV12*
PH1P - BC1X)
CH12 =
A1*C11*(SH12*
PH1P - BC1X)
CV11 =
A1*C11*(SV11*PH1
- BC1X)

27
CH11 =
A1*C11*(SH11*PH1
- BC1X)
CV2 =
A2*C2*(SV11*PH2 -
BC2X)
CH2 =
A2*C2*(SH11*PH2 -
BC2X)

28
CV4 =
A4*C4*(SV11*PH4 -
BC4X)
CH4 =
A4*C4*(SH11*PH4 -
BC4X)
CV9 =
A4*C9*(SV9*PH4 -
BC4X)

29
CH9 =
A4*C9*(SH9*PH4 -
BC4X)
CV1 =
A1*C2*PH1*C1PCV
CV3 =
A2*C4*PH2*C1PCV

/54.01
```

2

L-146

```
CHART TITLE - SUBROUTINE  TARGET(EVVR,EVVI,EHHR,EHHI)


  53.29--->*
           I           01                          THETA .GT. PI/2
 *---------------------*
 I    SV = CV12 +      I                             53.22--->*
 I   CV11 + CV2 +      I                               35    I  NOTE  12
 I CV4 + CV9 + CV1 +   I                            * * * * * * * * * * *
 I        CV3          I                            *      CONTINUE     *
 I                     I                            * * * * * * * * * * *
 I    SH = CH12 +      I                                      I
 I   CH11 + CH2 +      I                                      I
 I CH4 + CH9 - CV1 -   I                            RUCK-UFIMTSEV
 I        CV3          I                            SCATTERING
 *---------------------*                            (THETA.GT.90)
           I                                                  I
           I                                                  I
           I           02                                     I          13
 *---------------------*                            *---------------------*
 I SV = SV + CAD/SPI   I                            I        CV10 =       I
 I                     I                            I A4*C9*(PH4P*SV10    I
 I SH = SH + CAD/SPI   I                            I       - BC4X)       I
 *---------------------*                            I                     I
           I                                        I        CH10 =       I
           I                                        I A4*C9*(PH4P*SH10    I
           I           03                           I       - BC4X)       I
 *---------------------*                            I                     I
 I  COMPUTED GO TO     I                            I        CV9 =        I
 I   FOR  ISWA3        I                            I A4*C9*(PH4*SV9 -    I
 *---------------------*                            I        BC4X)        I
 I 40          54.18   I                            *---------------------*
 I 34          54.04   I                                      I
 *---------------------*                                      I
           I                                                  I          14
           I                                        *---------------------*
IF OUTSIDE THE RANGE                                I        CH9 =        I
           I                                        I A4*C9*(PH4*SH9 -    I
  54.03--->I                                        I        BC4X)        I
     34    I  NOTE  04                              I                     I
 * * * * * * * * * * *                              I        CV4 =        I
 *     CONTINUE      *                              I A4*C4*(PH4*SV11 -   I
 * * * * * * * * * * *                              I      BC4X)*Q4       I
           I                                        I                     I
           I                                        I        CH4 =        I
AZIMUTH NEAR ZERO,                                  I A4*C4*(PH4*SH11 -   I
ADD IN FAR EDGE TERMS                               I      BC4X)*Q4       I
WHICH ARE                                           *---------------------*
DIRECTLY ILLUMINATED                                          I
           I                                                  I
           I           05                                     I          15
 *---------------------*                            *---------------------*  *----------------->*
 I  XCS5 = AS5*AK2S    I                            I        CV2 =        I  I                 I          19
 I                     I                            I A2*C2*(PH2*SV11 -   I  I     *---------------------*
 I  XCS7 = AS7*AK2S    I                            I      BC2X)*Q2       I  I     I  SV = - SV*SPIK     I
 *---------------------*                            I                     I  I     I                     I
                                                    I        CH2 =        I  I     I  SH = SH*SPIK       I
```


CONTINUE
AZIMUTH NEAR ZERO, ADD IN FAR EDGE TERMS WHICH ARE DIRECTLY ILLUMINATED
05
XCS5 = AS5*AK2S
XCS7 = AS7*AK2S
06
7.0 1
FESL (XCS5,B50,B51, B52)
07
7.0 1
EESL (XCS7,E70,E71, E72)
08
PH5P = CMPLX(B50,B51)
PH7P = CMPLX(E70,E71)
CV6 = A2*C2*(PH2P*SV12 - BC2X)*Q6
09
CH6 = A2*C2*(PH2P*SH12 - BC2X)*Q6
CV8 = A4*C4*(PH4P*SV12 - BC4X)*Q8
CH8 = A4*C4*(PH4P*SH12 - BC4X)*Q8
10
CV5 = - AS5*C2*PH5P*CKV* Q6
CV7 = - AS7*C4*PH7P*CKV* Q9
SV = CV6 + CV8 + CV5 + CV7 + SV
11
SH = CH6 + CH8 - CV5 - CV7 + SH
CV4 = A4*C4*(PH4*SV11 - BC4X)*Q4
CH4 = A4*C4*(PH4*SH11 - BC4X)*Q4
15
CV2 = A2*C2*(PH2*SV11 - BC2X)*Q2
CH2 = A2*C2*(PH2*SH11 - BC2X)*Q2
CV11 = A1*C11*(PH1*SV11 - BC1X)*Q11
16
CH11 = A1*C11*(PH1*SH11 - BC1X)*Q11
CV1 = A1*C2*PH1*CK1*Q1
CV3 = A2*C4*PH2*CK1*Q3
17
SV = CV10 + CV9 + CV4 + CV2 + CV11 + CV1 + CV3
SH = CH10 + CH9 + CH4 + CH2 + CH11 - CV1 - CV3
54.03
40
NOTE 18
CONTINUE
19
SV = - SV*SPIK
SH = SH*SPIK
REFERENCE PHASE TO FRONT OF SECOND CYLINDER
20
SV = SV*C4
SH = SH*C4
21
EVVR(I) = REAL(SV)
EVVI(I) = - AIMAG(SV)
EHHR(I) = REAL(SH)
EHHI(I) = - AIMAG(SH)
30
22
END OF DO LOOP?
NO
53
10
YES
23
EXIT

2

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
             COMMON MOVER, M, NMIN, NMAX, DF, FC, PW, TC
             COMMON /TARS / ASPECT,ITT
             COMPLEX PH1, PH2, PH4, PH1P, PH2P, PH4P, C11, C2, C4, C9,
                     PH5P, PH7P
             COMPLEX  SV,SH
             COMPLEX CV12,CH12, CV11,CH11, CV2,CH2, CV4,CH4, CV9,CH9,
                     CV10,CH10, CV1, CV3, CV6,CH6, CV8,CH8, CV5, CV7
             COMPLEX CD1, CD2, CD3, CAD
             DIMENSION EVVR(1), EVVI(1), EHHR(1), EHHI(1)
   5001      FORMAT ( 7F8.0)
   6001      FORMAT ( '1  PROGRAM INPUT PARAMETERS',//,' THETA =',F9.3 )
   6002      FORMAT ( '0  A1 =', F7.4,'   H1 =',F7.4,/,
                     '  A2 =', F7.4,'   H2 =',F7.4,/,
                     '  A4 =', F7.4,'   H3 =',F7.4 )
   6005      FORMAT ( '  ALF1 = ',F8.3,/,'  ALF2 = ',F8.3,/,'  ALF3 = ',F8.3)
```

04/26/76   AUTOFLOW CHART SET - FWD/SCL RADSIM

CHART TITLE - FUNCTION Q(Z)

L-148

```
Q(Z) = 0.5*(1 +
ERF(Z))
 * ERF(Z) IS
EVALUATED USING A
RATIONAL POLYNOMIAL
APPROXIMATION
 * REFERENCE (HANDBK
MATH FUNCT BY
ABRAMOWITZ AND
STEGUN,
 *           SECTION
7.1.26)

          *
          |
          *   03
        *   *
      *       *
    *           *  TRUE
  *   Z .GT. 2.   *------------------------->*
    *           *                    10     |         09
      *       *                 *--------------------*
        *   *                   |      Q = 1.         |
          *                     *--------------------*
          |FALSE                            |
          |                                 |
          |                                 | 10
          |                           ---------
          |                          *  EXIT   *
          |                           ---------
          *   04
        *   *
      *       *
    *           *  TRUE
  *   Z .LT. - 2. *-----+
    *           *       |
      *       *         |
        *   *           +------------------>*
          *
```

2

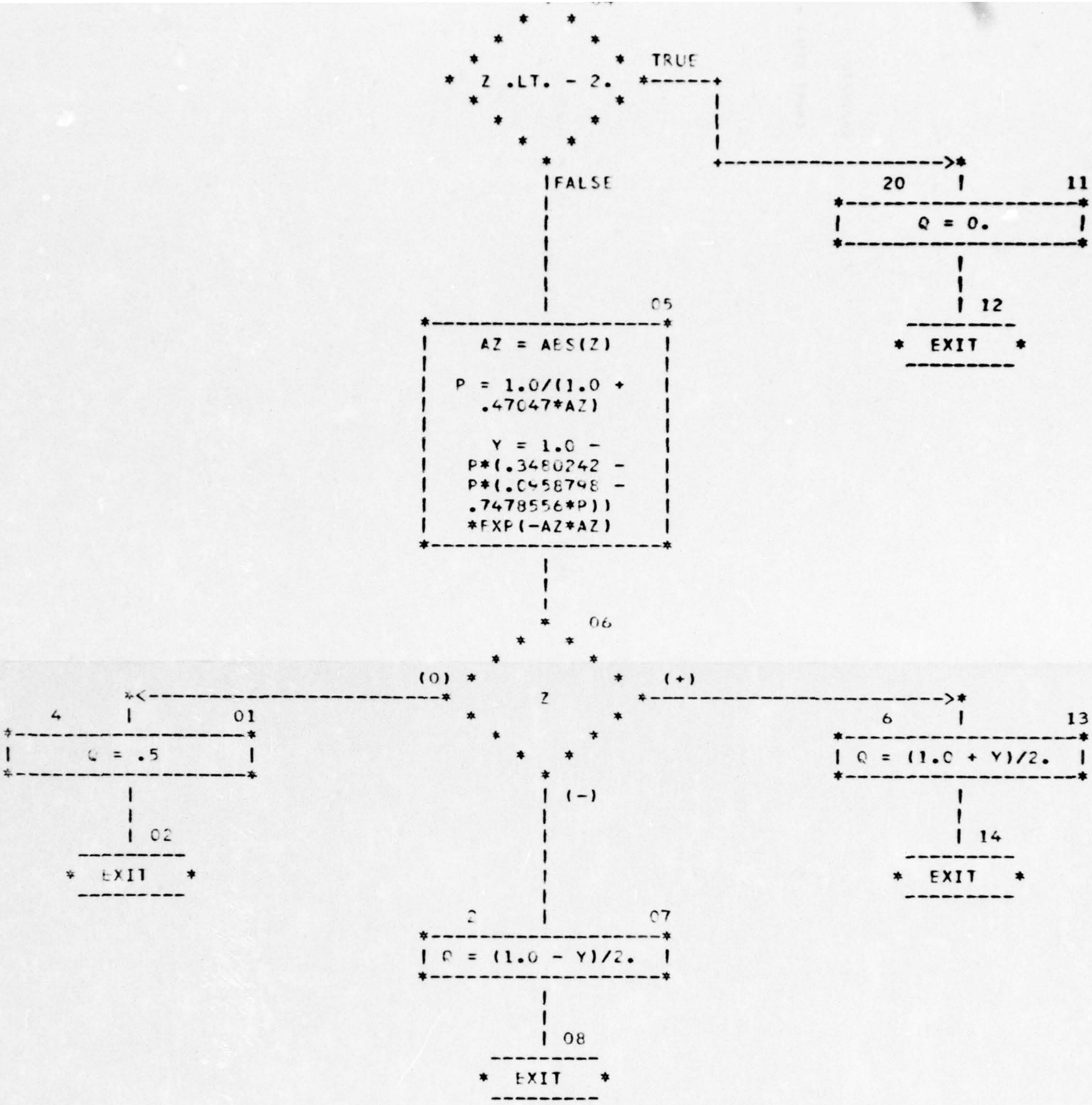
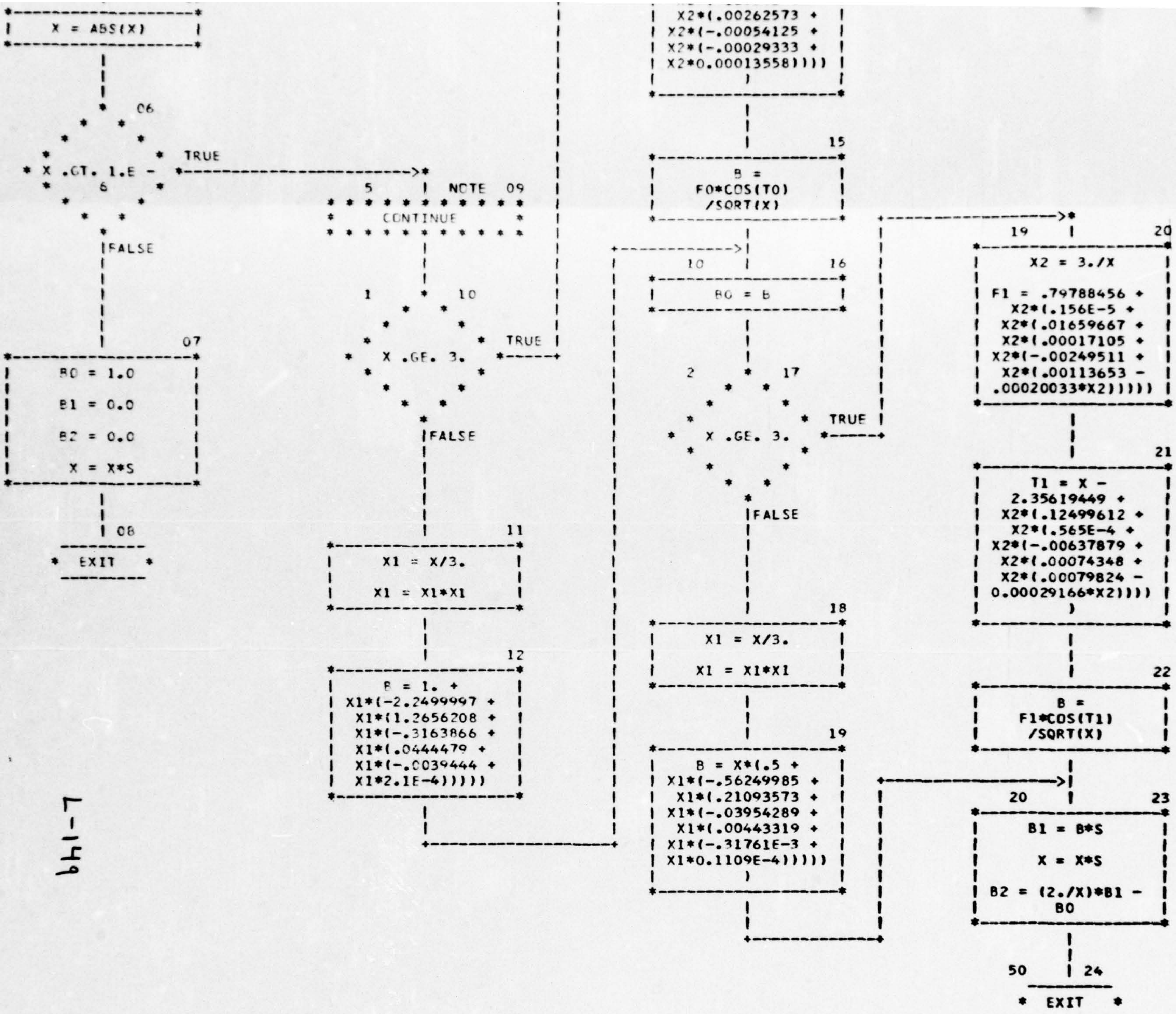

Z .LT. - 2.
TRUE
FALSE
20
11
Q = 0.
12
EXIT
05
AZ = ABS(Z)
P = 1.0/(1.0 + .47047*AZ)
Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P)) *EXP(-AZ*AZ)
06
Z
(0)
(+)
(-)
4
01
Q = .5
02
EXIT
6
13
Q = (1.0 + Y)/2.
14
EXIT
2
07
Q = (1.0 - Y)/2.
08
EXIT

CHART TITLE - SUBROUTINE BESL(X,FO,B1,B2)

```
         -----------
        /   BESL   /
         -----------

             *

     * BESSEL FUNCTION
    SUBROUTINE UTILIZING
    POLYNOMIAL
    APPROXIMATIONS
     * COMPUTES J0,J1,OR
    J2 FOR POSITIVE REAL
    ARGUMENTS
     * REFERENCE (HNDEK
    MATH FUNCT BY
    ABRAMOWITZ AND STEGUN
    SECTION 9.4 )

             *
             |        02
    *------------------*
    |     S = 1.0      |
    *------------------*
             |
             |
             |
             *    03                                       +----------------->*
           *   *                                           |            9     |       13
          *      *                                         |        *-------------------*
  FALSE *          *                                       |        |     X2 = 3./X     |
+-----*  X .LT. 0.0  *                                     |        |                   |
|       *          *                                       |        | FO = .79788456 +  |
|         *      *                                         |        |   X2*(-.77E-6 +   |
|          *   *                                           |        | X2*(-.00552740 +  |
|            *                                             |        |  X2*(-.9512E-4 +  |
|            |TRUE                                         |        |  X2*(.00137237 +  |
|            |                                             |        | X2*(-.72805E-3 +  |
|            |                                             |        | X2*0.14476E-3)))) |
|            |                                             |        |         )         |
|            |                                             |        *-------------------*
|            |                                             |                  |
|            |        04                                   |                  |
|   *------------------*                                   |                  |        14
|   |    S = - 1.0     |                                   |        *-------------------*
|   *------------------*                                   |        |     TO = X -      |
|                                                          |        |    .78539816 +    |
+----------->|                                             |        | X2*(-.04166397 +  |
             |        05                                   |        |  X2*(-.3954E-4 +  |
    *------------------*                                   |        |  X2*(.00262573 +  |
    |    X = ABS(X)    |                                   |        | X2*(-.00054125 +  |
    *------------------*                                   |        | X2*(-.00029333 +  |
             |                                             |        | X2*0.00013558)))) |
             |                                             |        |         )         |
             |                                             |        *-------------------*
             *    06                                       |                  |
```


X = ABS(X)
06
X .GT. 1.E - 6
TRUE
FALSE
07
B0 = 1.0
B1 = 0.0
B2 = 0.0
X = X*S
08
EXIT
5
NOTE 09
CONTINUE
1
10
X .GE. 3.
TRUE
FALSE
11
X1 = X/3.
X1 = X1*X1
12
B = 1. +
X1*(-2.2499997 +
X1*(1.2656208 +
X1*(-.3163866 +
X1*(.0444479 +
X1*(-.0039444 +
X1*2.1E-4)))))
X2*(.00262573 +
X2*(-.00054125 +
X2*(-.00029333 +
X2*0.00013558))))
)
15
B =
F0*COS(T0)
/SQRT(X)
10
16
B0 = B
2
17
X .GE. 3.
TRUE
FALSE
18
X1 = X/3.
X1 = X1*X1
19
B = X*(.5 +
X1*(-.56249985 +
X1*(.21093573 +
X1*(-.03954289 +
X1*(.00443319 +
X1*(-.31761E-3 +
X1*0.1109E-4))))))
)
19
20
X2 = 3./X
F1 = .79788456 +
X2*(.156E-5 +
X2*(.01659667 +
X2*(.00017105 +
X2*(-.00249511 +
X2*(.00113653 -
.00020033*X2)))))
21
T1 = X -
2.35619449 +
X2*(.12499612 +
X2*(.565E-4 +
X2*(-.00637879 +
X2*(.00074348 +
X2*(.00079824 -
0.00029166*X2))))
)
22
B =
F1*COS(T1)
/SQRT(X)
20
23
B1 = B*S
X = X*S
B2 = (2./X)*B1 -
B0
50
24
EXIT

L-149

```
04/26/76        INPUT LISTING                          AUTOFLOW CHART SET - FWD/SCL  RADSIM

  CARD NO        ****                           CONTENTS                                   ****

   1914                 TPIO2      = 3.*PIO2                                          RCS1 034
   1915                 DTR        = PI/180.                                          RCS1 035
   1916                 RTD        = 180./PI                                          RCS1 036
   1917                 DLT    = 15.*DTR                                              RCS1 037
   1918           C                                                                   RCS1 038
   1919                 READ (5,5001)          A1, A2, A4, H1, H2, H3                 RCS1 039
   1920            5001 FORMAT ( 7F8.0)                                               RCS1 040
   1921                 WRITE (6,6001) ASPECT                                         RCS1 042
   1922            6001 FORMAT ( '1  PROGRAM INPUT PARAMETERS',//,' THETA =',F9.3 )   RCS1 043
   1923                 WRITE (6,6002) A1,H1,A2,H2,A4,H3                              RCS1 044
   1924            6002 FORMAT ( '0  A1 =', F7.4,'   H1 =',F7.4,/,                    RCS1 045
   1925               1         '  A2 =', F7.4,'   H2 =',F7.4,/,                      RCS1 046
   1926               2         '  A4 =', F7.4,'   H3 =',F7.4 )                       RCS1 047
   1927           C                                                                   RCS1 048
   1928                 TH2        = H2+H2                                            RCS1 049
   1929                 TH2PH1     = TH2+H1                                           RCS1 050
   1930                 A21        = A2-A1                                            RCS1 051
   1931                 A41        = A4-A1                                            RCS1 052
   1932                 A42        = A4-A2                                            RCS1 053
   1933                 SA1    = SQRT(A1)                                             RCS1 054
   1934                 SA2    = SQRT(A2)                                             RCS1 055
```

```
      DLT    = 15.*DTR                                                   RCS1 037
C                                                                        RCS1 038
      READ (5,5001)          A1, A2, A4, H1, H2, H3                      RCS1 039
 5001 FORMAT ( 7F8.0)                                                    RCS1 040
      WRITE (6,6001) ASPECT                                              RCS1 042
 6001 FORMAT ( '1  PROGRAM INPUT PARAMETERS',//,' THETA =',F9.3 )        RCS1 043
      WRITE (6,6002) A1,H1,A2,H2,A4,H3                                   RCS1 044
 6002 FORMAT ( '0  A1 =', F7.4,'   H1 =',F7.4,/,                         RCS1 045
     1         '   A2 =', F7.4,'   H2 =',F7.4,/,                         RCS1 046
     2         '   A4 =', F7.4,'   H3 =',F7.4 )                          RCS1 047
C                                                                        RCS1 048
      TH2       = H2+H2                                                  RCS1 049
      TH2PH1    = TH2+H1                                                 RCS1 050
      A21       = A2-A1                                                  RCS1 051
      A41       = A4-A1                                                  RCS1 052
      A42       = A4-A2                                                  RCS1 053
      SA1   = SQRT(A1)                                                   RCS1 054
      SA2   = SQRT(A2)                                                   RCS1 055
C                                                                        RCS1 056
      ALF1     = ATAN(A21/H1)                                            RCS1 057
      ALF2     = ATAN(A42/TH2)                                           RCS1 058
      ALF3     = ATAN(A41/TH2PH1)                                        RCS1 059
C                                                                        RCS1 060
      PIMAL2   = PI-ALF2                                                 RCS1 061
      PIMAL3   = PI-ALF3                                                 RCS1 062
```

L-149a

2

1

L-1496

```
      C                                                                  RCS1 063
            X1D = ALF1*RTD                                               RCS1 064
            X2D = ALF2*RTD                                               RCS1 065
            X3D = ALF3*RTD                                               RCS1 066
      C                                                                  RCS1 067
            WRITE (6,6005) X1D,X2D,X3D                                   RCS1 068
       6005 FORMAT ( '  ALF1 = ',F8.3,/,'  ALF2 = ',F8.3,/,'  ALF3 = ',F8.3) RCS1 069
      C                                                                  RCS1 070
            TKFC    = FTK*FC*2.                                          RCS1 071
            XN      = 1.5                                                RCS1 072
            XNO2    = XN/2.                                              RCS1 073
            SOXN    = SIN(PI/XN)/XN                                      RCS1 074
            CXN     = -SOXN/XN                                           RCS1 075
            CPON    = COS(PI/XN)                                         RCS1 076
      C                                                                  RCS1 077
          6 CONTINUE                                                     RCS1 078
            TH         = ASPECT*DTR                                      RCS1 079
            STH        = SIN(TH)                                         RCS1 080
            CTH        = COS(TH)                                         RCS1 081
            TANTH      = STH/CTH                                         RCS1 082
            CK1        = 0.5*TANTH                                       RCS1 083
            CKV        = 0.5/TANTH                                       RCS1 084
            C1PCV = CK1+CKV                                              RCS1 085
      C                                                                  RCS1 086
            IF ( TH .GT. PIO2) GO TO 20                                  RCS1 087
      C
```

```
   1965         C                                                           RCS1 086
   1966               IF ( TH .GT. PIO2) GO TO 20                           RCS1 087
   1967         C                                                           RCS1 088
   1968               B12    = SCXN/(CPON - COS((PIO2-TH)/XNO2))           RCS1 089
   1969               B11    = SCXN/(CPON - COS((PIO2+TH)/XNO2))           RCS1 090
   1970               E9     = SCXN/(CPON - COS(TH/XNO2))                  RCS1 091
   1971         C     EDGE DIFFRACTION COEFFICIENTS (THETA.LT.PI/2)         RCS1 092




04/26/76      INPUT LISTING              AUTOFLOW CHART SET - FWD/SCL  RADSIM

  CARD NO       ****                     CONTENTS                      ****

   1972               SV12 = CXN-E12                                        RCS1 093
   1973               SH12 = CXN+E12                                        RCS1 094
   1974               SV11 = CXN-B11                                        RCS1 095
   1975               SH11 = CXN+B11                                        RCS1 096
   1976               SV9  = CXN-E9                                         RCS1 097
   1977               SH9  = CXN+E9                                         RCS1 098
   1978         C                                                           RCS1 099
   1979         C     EFFECTIVE AREA (DOUBLY REFLECTED ) TERMS(ADX)         RCS1 100
   1980               IF (TH-ALF1) 51,52,52                                 RCS1 101
   1981            51 AD1   = SA1*H1*STH                                    RCS1 102
   1982               GO TO 53                                              RCS1 103
   1983            52 AD1   = SA1*A21*CTH                                   RCS1 104
```

3

```
C     EFFECTIVE AREA (DOUBLY REFLECTED ) TERMS(ADX)                         RCS1 100
      IF (TH-ALF1) 51,52,52                                                 RCS1 101
   51 AD1   = SA1*H1*STH                                                    RCS1 102
      GO TO 53                                                              RCS1 103
   52 AD1   = SA1*A21*CTH                                                   RCS1 104
   53 CONTINUE                                                              RCS1 105
C                                                                           RCS1 106
      IF (TH-ALF2) 54,55,55                                                 RCS1 107
   54 AD2   = SA2*TH2*STH                                                   RCS1 108
      GO TO 56                                                              RCS1 109
   55 AD2   = SA2*A42*CTH                                                   RCS1 110
   56 CONTINUE                                                              RCS1 111
C                                                                           RCS1 112
      IF (TH-ALF1) 57,58,58                                                 RCS1 113
   58 IF (TH-ALF2) 59,59,57                                                 RCS1 114
   59 IF (TH-ALF3) 60,61,61                                                 RCS1 115
   60 AD3   = SA1*(H1*STH - A21*CTH)                                        RCS1 116
      GO TO 63                                                              RCS1 117
   61 AD3   = SA1*(A42*CTH - TH2*STH)                                       RCS1 118
      GO TO 63                                                              RCS1 119
   57 AD3 = 0                                                               RCS1 120
```

```
C      SMOOTHING FUNCTIONS                                          RCS1 154
       Q1   = Q(TKFC*A1*XALF)                                       RCS1 155
       Q2  = Q(TKFC*A2*(PIMAL2-TH))                                 RCS1 156
       Q3   = Q(TKFC*A2*XALF)                                       RCS1 157
       Q4  = Q(TKFC*A4*(PI-TH))                                     RCS1 158
       Q11 = Q(TKFC*A1*(PIMAL3-TH))                                 RCS1 159
    35 CONTINUE                                                     RCS1 160
C                                                                   RCS1 161
C      FREQUENCY LOOP                                               RCS1 162
C                                                                   RCS1 163
       DO 30 I = NMIN,NMAX                                          RCS1 164
       XI     = I-1                                                 RCS1 165
       AK     = XI*FTKDF                                            RCS1 166
       AK2    = AK+AK                                               RCS1 167
       AK2S   = AK2*STH                                             RCS1 168
C                                                                   RCS1 169
       XC1  = A1*AK2S                                               RCS1 170
       XC2  = A2*AK2S                                               RCS1 171
       XC4  = A4*AK2S                                               RCS1 172
C                                                                   RCS1 173
       CALL BESL(XC1,BJ10,BJ11,BJ12)                                RCS1 174
       CALL BESL(XC2,BJ20,BJ21,BJ22)                                RCS1 175
       CALL BESL(XC4,BJ40,BJ41,BJ42)                                RCS1 176
C                                                                   RCS1 177
       EC1  = BJ10+BJ12                                             RCS1 178
```

L-149c

L-149d

```
            BC2  = BJ20+BJ22                                           RCS1 179
            BC4  = BJ40+BJ42                                           RCS1 180
            BC1X = BC1*CXN                                             RCS1 181
            BC2X = BC2*CXN                                             RCS1 182
            BC4X = BC4*CXN                                             RCS1 183
C                                                                      RCS1 184
            PH1P = CMPLX(BJ10,BJ11)                                    RCS1 185
            PH2P = CMPLX(BJ20,BJ21)                                    RCS1 186
            PH4P = CMPLX(BJ40,BJ41)                                    RCS1 187
            PH1  = CONJG(PH1P)                                         RCS1 188
            PH2  = CONJG(PH2P)                                         RCS1 189
            PH4  = CONJG(PH4P)                                         RCS1 190
C                                                                      RCS1 191
            PS11 = -AK2*(H2+H1)*CTH                                    RCS1 192
            PS2  = -AK2*H2*CTH                                         RCS1 193
            PS9  =  AK2*(H2+H3)*CTH                                    RCS1 194
C                                                                      RCS1 195
            C11  = CMPLX(COS(PS11),SIN(PS11))                          RCS1 196
            C2   = CMPLX(COS(PS2),SIN(PS2))                            RCS1 197
            C4   = CONJG(C2)                                           RCS1 198
            C9   = CMPLX(COS(PS9),SIN(PS9))                            RCS1 199
C                                                                      RCS1 200
            IF ( TH .GT. PIO2 ) GO TO 35                               RCS1 201
C                                                                      RCS1 202
C           GENERAL REGION (THETA .GT. ALF3 .AND. THETA .LT. PI/2)     RCS1 203
```

```
 2081        C                                                              RCS1 202
 2082        C     GENERAL REGION (THETA .GT. ALF3 .AND. THETA .LT. PI/2)   RCS1 203
 2083        C                                                              RCS1 204
 2084        C     PHASE OF DOUBLY REFLECTED  SCATTERING TERMS              RCS1 205
 2085              PSD1 = -AK2*(A1*STH + H2*CTH)                            RCS1 206
 2086              PSD2 = -AK2*(A2*STH - H2*CTH)                            RCS1 207
 2087              PSD3 = -AK2*(A1*STH - H2*CTH)                            RCS1 208
```

```
04/26/76        INPUT LISTING                    AUTOFLOW CHART SET - FWD/SCL  RADSIM

 CARD NO       ****                        CONTENTS                             ****

 2088        C                                                              RCS1 209
 2089              CD1  = CMPLX(COS(PSD1),SIN(PSD1))                        RCS1 210
 2090              CD2  = CMPLX(COS(PSD2),SIN(PSD2))                        RCS1 211
 2091              CD3  = CMPLX(COS(PSD3),SIN(PSD3))                        RCS1 212
 2092        C                                                              RCS1 213
 2093        C     DOUBLY REFLECTED SCATTERING (TOTAL)                      RCS1 214
 2094              CAF  = SQRT(AK2*AK2)*(AD1*CD1 + AD2*CD2 + AD3*CD3)       RCS1 215
 2095        C                                                              RCS1 216
 2096        C     PUCK-UFIMTSEV SCATTERING (THETA.LT.90,EDGES ILLUMINATED AT 90)   RCS1 217
 2097              CVI2 = A1*CI1*(SVI2*PHIP-BC1X)                           RCS1 218
 2098              CHI2 = A1*CI1*(SHI2*PHIP-FC1X)                           RCS1 219
 2099              CVI1 = A1*CI1*(SVI1*PHI-EC1X)                            RCS1 220
 2100              CHI1 = A1*CI1*(SHI1*PHI-FC1X)                            RCS1 221
```

3

```
      C                                                                RCS1 216
      C     BUCK-UFIMTSEV SCATTERING (THETA.LT.90,EDGES ILLUMINATED AT 90)   RCS1 217
            CV12 = A1*C11*(SV12*PH1P-BC1X)                             RCS1 218
            CH12 = A1*C11*(SH12*PH1P-BC1X)                             RCS1 219
            CV11 = A1*C11*(SV11*PH1-BC1X)                              RCS1 220
            CH11 = A1*C11*(SH11*PH1-BC1X)                              RCS1 221
            CV2  = A2*C2*(SV11*PH2-BC2X)                               RCS1 222
            CH2  = A2*C2*(SH11*PH2-BC2X)                               RCS1 223
            CV4  = A4*C4*(SV11*PH4-BC4X)                               RCS1 224
            CH4  = A4*C4*(SH11*PH4-BC4X)                               RCS1 225
            CV9  = A4*C9*(SV9*PH4-BC4X)                                RCS1 226
            CH9  = A4*C9*(SH9*PH4-BC4X)                                RCS1 227
            CV1  = A1*C2*PH1*C1PCV                                     RCS1 228
            CV3  = A2*C4*PH2*C1PCV                                     RCS1 229
            SV   = CV12+CV11+CV2+CV4+CV9 +CV1 +CV3                     RCS1 230
            SH   = CH12+CH11+CH2+CH4+CH9 -CV1 -CV3                     RCS1 231
      C                                                                RCS1 232
            SV   = SV + CAD/SPI                                        RCS1 233
            SH   = SH + CAD/SPI                                        RCS1 234
      C                                                                RCS1 235
            GO TO (40,34), ISWA2                                       RCS1 236
```

```
   34 CONTINUE                                                          RCS1 237
C     AZIMUTH NEAR ZERO, ADD IN FAR EDGE TERMS WHICH ARE                RCS1 238
C           DIRECTLY ILLUMINATED                                        RCS1 239
      XCS5 = AS5*AK25                                                   RCS1 240
      XCS7 = AS7*AK25                                                   RCS1 241
      CALL BESL(XCS5,B50,B51,B52)                                       RCS1 242
      CALL BESL(XCS7,B70,B71,B72)                                       RCS1 243
      PH5P = CMPLX(B50,B51)                                             RCS1 244
      PH7P = CMPLX(B70,B71)                                             RCS1 245
      CV6  = A2*C2*(PH2P*SV12-BC2X)*Q6                                  RCS1 246
      CH6  = A2*C2*(PH2P*SH12-BC2X)*Q6                                  RCS1 247
      CV8  = A4*C4*(PH4P*SV12-BC4X)*Q8                                  RCS1 248
      CH8  = A4*C4*(PH4P*SH12-BC4X)*Q8                                  RCS1 249
      CV5  =-AS5*C2*PH5P*CKV*Q6                                         RCS1 250
      CV7  =-AS7*C4*PH7P*CKV*Q8                                         RCS1 251
      SV   = CV6 + CV8 +CV5+CV7 + SV                                    RCS1 252
      SH   = CH6 + CH8 -CV5-CV7 + SH                                    RCS1 253
      GO TO 40                                                          RCS1 254
C                                                                       RCS1 255
C     THETA .GT. PI/2                                                   RCS1 256
   35 CONTINUE                                                          RCS1 257
C                                                                       RCS1 258
C     RUCK-UFIMTSEV SCATTERING (THETA.GT.90)                            RCS1 259
      CV10 = A4*C9*(PH4P*SV10-BC4X)                                     RCS1 260
      CH10 = A4*C9*(PH4P*SH10-BC4X)                                     RCS1 261
   2139             CV10 = A4*C9*(PH4P*SV10-BC4X)                              RCS1 260
   2140             CH10 = A4*C9*(PH4P*SH10-BC4X)                              RCS1 261
   2141             CV9  = A4*C9*(PH4*SV9-BC4X)                                RCS1 262
   2142             CH9  = A4*C9*(PH4*SH9-BC4X)                                RCS1 263
   2143             CV4  = A4*C4*(PH4*SV11-BC4X)*Q4                            RCS1 264
   2144             CH4  = A4*C4*(PH4*SH11-BC4X)*Q4                            RCS1 265
   2145             CV2  = A2*C2*(PH2*SV11-BC2X)*Q2                            RCS1 266
```

```
04/26/76        INPUT LISTING                 AUTOFLOW CHART SET - FWD/SCL  RADSIM

  CARD NO         ****                      CONTENTS                           ****

   2146             CH2  = A2*C2*(PH2*SH11-BC2X)*Q2                            RCS1 267
   2147             CV11 = A1*C11*(PH1*SV11-BC1X)*Q11                          RCS1 268
   2148             CH11 = A1*C11*(PH1*SH11-BC1X)*Q11                          RCS1 269
   2149             CV1  = A1*C2*PH1*CK1*Q1                                    RCS1 270
   2150             CV3  = A2*C4*PH2*CK1*Q3                                    RCS1 271
   2151             SV   = CV10+CV9+CV4+CV2+CV11 +CV1+CV3                      RCS1 272
   2152             SH   = CH10+CH9+CH4+CH2+CH11 -CV1-CV3                      RCS1 273
   2153          40 CONTINUE                                                   RCS1 274
   2154        C                                                               RCS1 275
   2155             SV =-SV*SPIK                                               RCS1 276
   2156             SH = SH*SPIK                                               RCS1 277
   2157        C                                                               RCS1 278
   2158        C    REFERENCE PHASE TO FRONT OF SECOND CYLINDER
```

```
CARD NO      ****                         CONTENTS                                ****

  2146              CH2  = A2*C2*(PH2*SH11-RC2X)*Q2                            RCS1 267
  2147              CV11 = A1*C11*(PH1*SV11-RC1X)*Q11                          RCS1 268
  2148              CH11 = A1*C11*(PH1*SH11-RC1X)*Q11                          RCS1 269
  2149              CV1  = A1*C2*PH1*CK1*Q1                                    RCS1 270
  2150              CV3  = A2*C4*PH2*CK1*Q3                                    RCS1 271
  2151              SV   = CV10+CV9+CV4+CV2+CV11 +CV1+CV3                      RCS1 272
  2152              SH   = CH10+CH9+CH4+CH2+CH11 -CV1-CV3                      RCS1 273
  2153           40 CONTINUE                                                   RCS1 274
  2154         C                                                               RCS1 275
  2155              SV =-SV*SPIK                                               RCS1 276
  2156              SH = SH*SPIK                                               RCS1 277
  2157         C                                                               RCS1 278
  2158         C    REFERENCE PHASE TO FRONT OF SECOND CYLINDER                RCS1 279
  2159         C                                                               RCS1 280
  2160              SV   = SV*C4                                               RCS1 281
  2161              SH   = SH*C4                                               RCS1 282
  2162         C                                                               RCS1 283
  2163              EVVR(I) = REAL(SV)                                         RCS1 284
  2164              EVVI(I) =-AIMAG(SV)                                        RCS1 285
  2165              EHHR(I) = REAL(SH)                                         RCS1 286
  2166              EHHI(I) =-AIMAG(SH)                                        RCS1 287
  2167           30 CONTINUE                                                   RCS1 288
  2168              RETURN                                                     RCS1 289
  2169              END                                                        RCS1 290
```

3

L-149e

```
      FUNCTION Q(Z)                                                   RCS1 291
C     Q(Z) = 0.5*(1 + ERF(Z))                                         RCS1 292
C   * ERF(Z) IS EVALUATED USING A RATIONAL POLYNOMIAL APPROXIMATION   RCS1 293
C   * REFERENCE (HANDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN,          RCS1 294
C   *             SECTION 7.1.26)                                     RCS1 295
C                                                                     RCS1 296
      IF ( Z.GT. 2.) GO TO 10                                         RCS1 297
      IF ( Z.LT.-2.) GO TO 20                                         RCS1 298
      AZ = ABS(Z)                                                     RCS1 299
      P = 1.0/(1.0 + .47047*AZ)                                       RCS1 300
      Y = 1.0 - P*(.3480242 - P*(.0958798 - .7478556*P))*EXP(-AZ*AZ)  RCS1 301
      IF (Z) 2,4,6                                                    RCS1 302
    2 Q = (1.0 - Y)/2.                                                RCS1 303
      RETURN                                                          RCS1 304
    4 Q = .5                                                          RCS1 305
      RETURN                                                          RCS1 306
    6 Q = (1.0 + Y)/2.                                                RCS1 307
      RETURN                                                          RCS1 308
   10 Q = 1.                                                          RCS1 309
      RETURN                                                          RCS1 310
   20 Q = 0.                                                          RCS1 311
      RETURN                                                          RCS1 312
      END                                                             RCS1 313
```

L-149 f

```
      SUBROUTINE BESL ( X, B0, B1, B2 )                                 RCS1 314
C                                                                       RCS1 315
C    * BESSEL FUNCTION SUBROUTINE UTILIZING POLYNOMIAL APPROXIMATIONS    RCS1 316
C    * COMPUTES J0,J1,OR J2 FOR POSITIVE REAL ARGUMENTS                  RCS1 317
C    * REFERENCE (HNDBK MATH FUNCT BY ABRAMOWITZ AND STEGUN SECTION 9.4 )RCS1 318
C                                                                       RCS1 319
      S = 1.0                                                           RCS1 320
      IF (X .LT. 0.0) S=-1.0                                            RCS1 321
      X = ABS(X)                                                        RCS1 322
C                                                                       RCS1 323
      IF ( X .GT. 1.E-6 ) GO TO 5                                       RCS1 324
      B0 = 1.0                                                          RCS1 325
      B1 = 0.0                                                          RCS1 326
      B2 = 0.0                                                          RCS1 327
      X = X * S                                                         RCS1 328
      RETURN                                                            RCS1 329
C                                                                       RCS1 330
    5 CONTINUE                                                          RCS1 331
C                                                                       RCS1 332
    1 IF ( X .GE. 3.) GO TO 9                                           RCS1 333
      X1 = X/3.                                                         RCS1 334
      X1 = X1*X1                                                        RCS1 335
      B0 = 1.+ X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479RCS1 336
     1 +      X1*(-.0039444+ X1*2.1E-4 )))) )                           RCS1 337
      GO TO 10                                                          RCS1 338
```

2

```
      X1 = X/3.                                                         RCS1 334
      X1 = X1*X1                                                        RCS1 335
      B = 1.+ X1*(-2.2499997+ X1*(1.2656208+ X1*(-.3163866+ X1*(.0444479RCS1 336
     1 +      X1*(-.0039444+ X1*2.1E-4 )))) )                           RCS1 337
      GO TO 10                                                          RCS1 338
C                                                                       RCS1 339
    9 X2 = 3./X                                                         RCS1 340
      F0 =  .79788456 +X2*(-.77E-6 +X2*(-.00552740 +X2*(-.9512E-4 +X2*  RCS1 341
     1     (.00137237 +X2*(-.72805E-3 +X2*0.14476E-3 )))) )             RCS1 342
      T0 = X - .78539816 +X2*(-.04166397 +X2*(-.39545E-4 +X2*(.00262573 RCS1 343
     1    +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))          RCS1 344
      B = F0*COS(T0)/SQRT(X)                                            RCS1 345
C                                                                       RCS1 346
   10 S0 = B                                                            RCS1 347
C                                                                       RCS1 348
    2 IF ( X .GE. 3. ) GO TO 19                                         RCS1 349
      X1 = X/3.                                                         RCS1 350
      X1 = X1*X1                                                        RCS1 351
      B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1*    RCS1 352
     1              (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4))))) )     RCS1 353
      GO TO 20                                                          RCS1 354
C                                                                       RCS1 355
   19 X2 = 3./X                                                         RCS1 356
      F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2*    RCS1 357
     1     (-.00249511 +X2*(.00113653 -.00020033*X2 )))))               RCS1 358
      T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879   RCS1 359
     1    +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))            RCS1 360
      B = F1*COS(T1)/SQRT(X)                                            RCS1 361
     1    +X2*(-.00054125 +X2*(-.00029333 +X2*0.00013558 )))))           RCS1 344
          B = F0*COS(T0)/SQRT(X)                                         RCS1 345
C                                                                        RCS1 346
       10 B0 = B                                                         RCS1 347
C                                                                        RCS1 348
        2 IF ( X .GE. 3. ) GO TO 19                                      RCS1 349
          X1 = X/3.                                                      RCS1 350
          X1 = X1*X1                                                     RCS1 351
          B = X*( .5 +X1*(-.56249985 +X1*(.21093573 +X1*(-.03954289 +X1* RCS1 352
     1                (.00443319 +X1*(-.31761E-3 +X1*0.1109E-4)))))) )   RCS1 353
          GO TO 20                                                       RCS1 354
C                                                                        RCS1 355
       19 X2 = 3./X                                                      RCS1 356
          F1 = .79788456 +X2*(.156E-5 +X2*(.01659667 +X2*(.00017105 +X2* RCS1 357
     1       (-.00249511 +X2*(.00113653 -.00020033*X2 )))))              RCS1 358
          T1 = X - 2.35619449 +X2*(.12499612 +X2*(.565E-4 +X2*(-.00637879 RCS1 359
     1    +X2*(.00074348 +X2*(.00079824 -0.00029166*X2 )))))             RCS1 360
          B = F1*COS(T1)/SQRT(X)                                         RCS1 361
C                                                                        RCS1 362
       20 B1 = B * S                                                     RCS1 363
          X = X * S                                                      RCS1 364
          B2= (2./X)*B1 - B0                                             RCS1 365
       50 RETURN                                                         RCS1 366
          END                                                            RCS1 367
```

L-1498 3

L.8 REFERENCES

1. Liang, C.S. and R. W. Clay, Improved Wideband Scattering Analysis, RADC-TR-73-11, (Rome Air Development Center, Rome, N.Y., January, 1973) pp. 29-52 AD#758245.

2. Ibid, pp. 159-162

3. Ibid, pp. A1-B7

4. Ibid, pp. C1-C15

5. Hong, S., S. L. Borrison, and D. P. Ford, "Short Pulse Scattering by Long Wire", IEEE Trans. Ant. Prop., Vol. AP-16, No. 3, pp. 338-339

6. Liang, Op. Cit., pp. 81-110

7. Ibid, pp. 53-80

8. This reference was intentionally omitted.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL
1776-1976